TOWARDS PRACTICAL IMPLEMENTATION OF COMPUTATIONAL SOLUTION OF THE KINEMATIC-WAVE MODEL FOR SIMULATING TRAFFIC-FLOW SCENARIOS

A Thesis

by

NISHANT KUMAR

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2004

Major Subject: Computer Science

TOWARDS PRACTICAL IMPLEMENTATION OF COMPUTATIONAL SOLUTION OF THE KINEMATIC-WAVE MODEL FOR SIMULATING TRAFFIC-FLOW SCENARIOS

A Thesis

by

NISHANT KUMAR

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Paul Nelson (Chair of Committee)	Riccardo Bettati (Member)
Peter Howard (Member)	Valerie Taylor (Head of Department)

August 2004

Major Subject: Computer Science

ABSTRACT

Towards Practical Implementation of Computational Solution of the Kinematic-wave Model for Simulating Traffic-flow Scenarios. (August 2004)

Nishant Kumar, B.Tech., Indian Institute of Technology, Bombay

Chair of Advisory Committee: Dr. Paul Nelson

The Kinematic-wave model is one of the models proposed to simulate vehicular traffic. It has not received widespread use because of poor understanding of associated interface conditions and early use of incorrect numerical schemes used. This thesis analyzes mathematically correct boundary and interface conditions in the context of the Godunov method as the numerical scheme for the simulation software created. This thesis simulates a set of scenarios originally proposed by Ross, to verify the validity of simulation. The results of the simulation are compared against the corresponding results of Ross, and against intuitive expectation of the behavior of actual traffic under the scenarios. Our results tend either to agree with or improve upon those reported by Ross, who used alternate models.

.

ACKNOWLEDGEMENTS

I would like to give my sincere thanks to Dr. Paul Nelson, whose guidance and constant support were instrumental in deciding the thesis topic and writing this thesis.

.

TABLE OF CONTENTS

.

CHAPTER Page

.

LIST OF FIGURES

.

Page

.

LIST OF TABLES

.

CHAPTER I

INTRODUCTION

It has always been a dream to be able to drive to ones destination through the traffic within the least time possible. To make this dream come true, it is helpful to be able to predict the traffic behavior, given any situation. This is the primary motivation behind the efforts put into modeling traffic. Modeling traffic has always been a very challenging task, because of the innumerable scenarios and their associated complexities. For example, the scenario could include modeling the traffic behavior at a signal, it could include modeling the traffic behavior during an accident when one or more lanes are blocked, or it might even be required to model lane merges, lane forking and various other complex forms of driver behavior. It is difficult to devise a traffic model that can simulate all scenarios with a reasonable degree of accuracy. It is perhaps more reasonable to talk about devising a model that can simulate a certain subset of traffic scenarios accurately. Hence, the choice of model now becomes dependent on the traffic scenarios being modeled. The Kinematic-wave model is one such model which tries to model traffic scenarios. For the present discussion, we have divided models into two categories: 1. Kinematic-wave Model and, 2. Alternate models.

1.1 Kinematic-wave Models of Traffic Flow

1.1.1 Description of Kinematic-wave Model

The Kinematic-wave Model (KWM) of traffic flow was proposed by Lighthill and Whitham [1955] and independently by Richards [1956]. The KWM is a simple continuum model, which in turn belongs to a class of traffic model termed as *continuum flow models*. Continuum flow models treat vehicular traffic somewhat as a one-dimensional compressible fluid [Kuhne et al. 2002]. This treatment leads to two fundamental equations. The first equation is the *law of conservation of vehicles*.

This thesis follows the style and format of *ACM Transactions on Mathematical Software*.

Consider a section of road at any time t. At this time there are no vehicles in the section being considered but at this time vehicles start entering this section. If we take a snapshot of the conditions at any time, total number of vehicles that entered the section should be same as the sum of total number of vehicles that left the section and number of vehicles remaining in the section. More specifically, conservation of vehicles can be written in differential form as

$$\frac{\partial k}{\partial t}+\frac{\partial q}{\partial x}=S(x,t). \tag{1}$$

Here:

$k=k(x,t)$ refers to the vehicular density. The vehicular density is defined as number of vehicles per unit length along the roadway at position x and time t.

$q=q(x,t)$ refers to the flow of vehicles, which can be defined as number of vehicles passing the position x per unit time, at time t;.

$S(x,t)$ refers to the rate per unit length at which vehicles enter (leave) the roadway, which includes the effect due to on-ramps/ off-ramps;

∂ refers to partial differentiation with respect to the specified variable.

The second fundamental equation of the KWM states that there is a deterministic relationship between speed and density or flow and density [Kuhne et al. 2002]. This means that flow at any time t and position x from the origin, can be expressed as a function of the density of vehicles at that point, and possibly also depends explicitly on the position or time. This relationship is expressed through the means of a traffic stream model (TSM). (TSMs are alternately known as fundamental diagram, or FDs, but we shall prefer the former). TSMs were first introduced by Greenshields [1934]. A generic TSM can be written as

$$q(x, \tau) = Q(k(x, \tau),x,t) \tag{2}$$

Here Q is some known function that relates q and k at a particular position x and time t.

1.1.2 Relation to Demand-Capacity Analysis

TSMs are often assumed to have decreasing derivatives with respect to k (i.e., being strictly concave in k). This means that,

$$\frac{\partial^2 Q}{\partial k^2} < 0\text{, on some interval, } \quad 0 \le k \le k_{jam} = k_{jam}(x)$$

on which $Q \geq 0$, and $Q \equiv 0$ for all values of k.

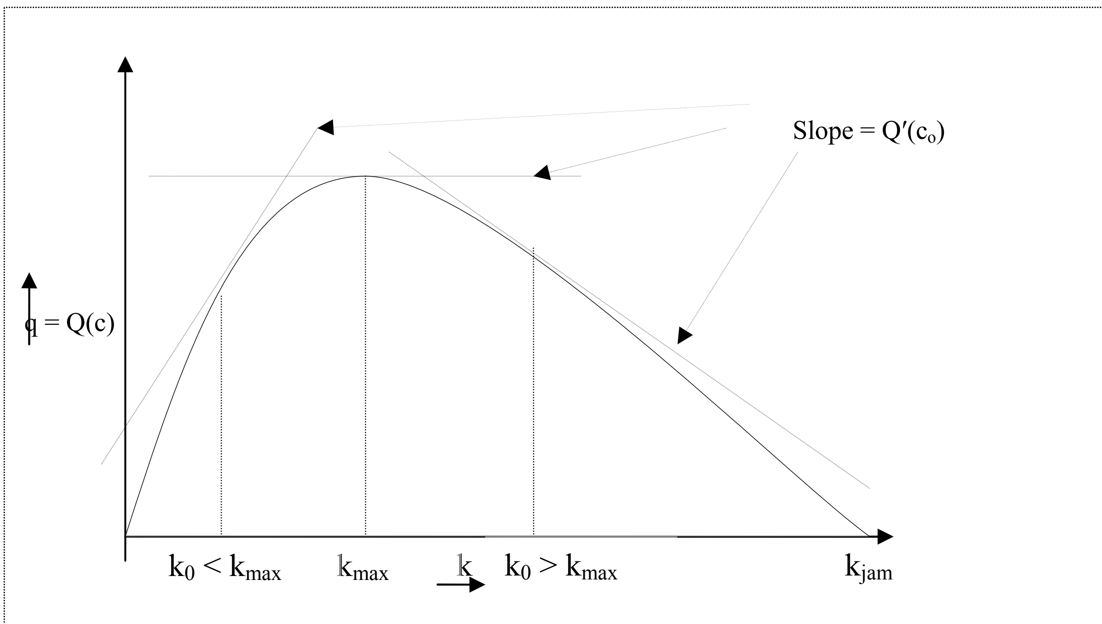


Figure 1. A typical flow–density graph (Traffic Stream Model)

This condition implies that the TSM predicts that there is a unique maximum flow at each x. This can be represented as $Q(k_{max}(x), x) = Q_{cap}(x)$, (Figure 1) where k_{max} is the density/ concentration at which the maximum flow, Q_{cap}, occurs. This value $Q_{cap} = Q_{cap}(x)$ is also known as the *capacity* of the roadway at location x. Capacity, as just defined, is one of the key elements of capacity-demand analysis, which is more or less the study of steady state solutions of the KWM.

1.1.3 Nature of Solutions

This subsection is condensation of thoughts of Lighthill and Whitham [1955] and of Richards [1956] regarding the nature of the solutions of the KWM. A fundamental hypothesis of the KWM is that at any point of road, the flow of vehicles is a function of the concentration of vehicles at that section, per the TSM (2). In the TSM, flow is defined as number of vehicles that pass through a fixed observation point per unit time, while concentration is defined as vehicles per unit distance at a particular point. The concentration can be per lane or it can be aggregated across lanes. Locally (i.e., at sufficiently short times) the flow develops as "kinematic waves" of constant density (and therefore flow) that propagate along the roadway at speed equal to the partial derivative of Q with respect to k; the latter is therefore known as "wave speed." This wave speed is different from the *vehicle speed* which can be graphically interpreted as slope of line between the origin and the point $(k, Q(k))$ on a TSM. Thus, for a strictly convex TSM, wave speed ≤ vehicle speed. Also, densities less than $k_{cap}(x)$ propagate downstream, and those greater than $k_{cap}(x)$ propagate upstream.

However, these kinematic waves can either intersect or diverge. The former produces a "shock wave," which is to say a discontinuity that propagates, most often (but not necessarily) upstream, with a speed determined by the densities immediately upstream and downstream as necessary to conserve vehicles ("shock condition"). An emerging shock wave is seen when the traffic slows down in response to the slowdown of traffic ahead. A driver notices it when he adjusts his speed due to the fluctuation in the speed of the car in front of him. This is apparent visually in the propagation of brake lights in a dense traffic situation.

Divergence of kinematic waves occurs when the wave speed is discontinuous and higher downstream than upstream (and therefore the density is discontinuous and higher upstream than downstream). This leads to so-called similarity solutions, in which the vehicle speed is necessarily higher downstream than upstream. An example of such "queue discharges" (also known in the traffic context as "starting waves", or "acceleration waves") is seen in the case that two lanes suddenly fork into three or more lanes. This distributes the traffic and hence increases the flow. It is also apparent visually when a signal changes to green.

Some more modern views of the nature of solutions of the KWM are discussed below, in conjunction with considerations of computational solution. We note that Newell [1995] has proposed an alternate, but completely equivalent, version of the KWM, within which flow and density are replaced as dependent variables by a single "cumulative flow." This formulation appears to have potential advantages, especially for circumstances in which it is truly the cumulative flow over an interval of time that is of dominant interest.

1.1.4 Issues

The KWM was one of the earliest models intended to predict traffic flow but till this date its acceptance in practical world is very limited. There are many reasons for this, and we now turn to a discussion of those. One notable exception to this is France [Daganzo 1995], where the KWM is used extensively in practice.

1.1.4.1 Continuum Description

The law of conservation of vehicles was first introduced in traffic flow, in 1955, by Lighthill and Whitham [1955], and independently in 1956 by Richards [1956]. There is little dispute about the *traffic conservation* equation (1), and it is widely accepted. In fact, Papageorgiou [1998] describes it as "the only accurate physical law available for traffic flow." Nonetheless, it is questionable to the extent that it idealizes traffic flow and density as continuous functions of position and time, whereas in fact in any individual instance these are necessarily discrete integer-value functions. This idealization is the defining characteristic of "continuum" models of traffic flow, of which the KWM is but one instance; other instances are briefly discussed in the following subsection.

Note however that equations for *conservation of vehicles* and TSMs are of a fundamentally different nature. The conservation law is an idealized mathematical version of the physically fundamental identity that the number of vehicles on a segment of roadway can change only through vehicles entering or leaving that segment of roadway. By contrast, TSMs are a hypothesis; although they are not an unreasonable hypothesis, the extent of their validity has been challenged on empirical grounds. Within the framework of this thesis we shall fundamentally assume validity of (1) and (2). Nonetheless, we shall now present a brief overview of the basis for prevailing doubts about TSMs.

1.1.4.2 Observational Data

A TSM translates to the hypothesis that drivers tend to adjust their speed according to density. As the latter is related to the spatial headway (distance between vehicles), this is intuitively appealing. Nonetheless, the hypothesized relationship between flow and vehicular density has been the source of many doubts. In fact Greenshields [1936] himself noted the following, in regard to obtaining observational data to determine an TSM: "At greater densities there is less consistency owing to the fact that a few slow moving vehicles retard the whole traffic stream. This means that the data for the higher density must be ample." Likewise Lighthill and Whitham [1955] observe that "certain measurements of traffic stopped and started indicate that under these conditions the mean concentration may be far less, and the mean speed far greater, than the values taken from the flow-concentration curve under more nearly steady circumstances." Both of these doubts from the seminal investigators have been supported by numerous subsequent observational studies. Most notably, Drake, Schofer and May [1967] found very substantial scatter in density/flow data at densities above about 35 vehicles/mile/lane, accompanied by a possible discontinuity (so-called "two-regime" flow) in the vicinity of capacity density. Likewise, Koshi [1963] observed both similar scatter, and the tendency for density/flow data to form an "inverted lambda" shape, again suggestive somehow of two different flow regimes in the vicinity of capacity density.

1.1.4.3 Deterministic vs. Statistical Interpretations and Filtering

TSM seems to suggest that there exists a deterministic relationship between flow and density, but this is not supported by observational data. This conflict of theoretical and observational data was used by Ross [1988] to try and invalidate KWM as a reliable traffic model. This claim was rebutted by Newell [1989] who argued that the relationship between speed and density is close to being determinate if their mean values are taken into account instead of values at a specific instance of time. We can take two different approaches to justify the use of TSM in spite of the observed data:

1. The first approach is data selection. We can leave remove the flow data which are too scattered, as they are more likely to be fluctuation limited to a specific instant. Once this is done, the remaining flow point would be closely clustered together. The definition of the curve that would most closely define this cluster on the density-flow plane can be selected as the TSM for the model.

2. The next approach is to modify the definition of TSM. Instead of interpreting TSM as deterministic relation between specific value of density and flow at any given instant, we can interpret it as a "relation between *mean* values over some suitably defined statistical distribution, of flow and density. In this approach the large scatter for congested flows would be simply a region within which the underlying distribution has a relatively large standard deviation." [Nelson, 2004]

Cassidy [1998] has followed the first of these approaches, and has concluded, "by plotting average values of the data corresponding to each nearly stationary condition, well-defined relations are observed."

As regards the second of these approaches, one clearly could obtain *some* functional relationship between mean values of density and flow, over some sufficiently large set of samples. Any departures from this relationship in any individual case could then be considered a "statistical fluctuation." According to Newell, there are huge variations in speed-density relationships when the number of vehicles under observation is in the range of 10-100 vehicles [Newell 1989]. This variation reduces when the observation comprises of averaging the values over a very large number of vehicles. In that case, the relationship is fairly predictable, but the former numbers of vehicle are more typical of the few-minute aggregation periods commonly used to collect data. Further in this regard, Muñoz and Daganzo [2002] suggest, from analysis of data gathered upstream of a small off-ramp with a tendency to over-saturation (demand exceeding capacity), that "the scatter in the flow-density data observed at our site can be explained for the entire observation period by statistical effects … and by the passage of the transition zone" (i.e., the interior of a shock wave, or region of transition from free flow to congested flow, that occurs just upstream of a developing queue).

1.1.4.4 Numerical Solution

There also is some apparent residual bias toward the KWM based on early use of computational methods that are incorrect [Newell 1989], in the sense that they converge to incorrect solutions. This is exacerbated by the fact that the KWM (1), (2) is not well-posed, even with suitable initial and boundary conditions, in the sense that multiple solutions can exist. It is necessary to impose an additional condition, the so-called entropy condition [Michalopoulos et al. 1984], to select the solution that correctly reflects the ability of drivers to anticipate conditions on the roadway ahead of them. A correct numerical scheme must then converge to the *entropy solution* picked out by this condition, and some simple and therefore intuitively appealing computational methods fail this test.

1.1.4.5 Boundary Conditions and Recent Understandings

Apart from the above stated concerns, there were lots of concerns regarding the representation of important concepts like boundaries, points of spatial discontinuity and point constrictions. Poor understanding of these concepts until recently was one of the main reasons for KWM model not achieving wider practical use.

With the recent advances in understanding of interface conditions and bottlenecks, coupled with the fact that the computational method used in KWM model was incorrect, there exist enough reasons to revisit the model. Michalopoulos, Beskos and Lin [1984] discuss boundary conditions, as do Daganzo [1995] and Lebacque [2003]. They describe boundary conditions appropriate to the KWM, as well as the appropriate (continuous and discrete) formulation of interface conditions. Lebacque [1996] has very recently shown that these boundary conditions are the specialization of the mathematically general boundary conditions for hyperbolic partial differential equations.

We now switch from a discussion of the KWM, to a broader discussion of quantitative models of traffic flow, but still within the framework of continuum models. Continuum alternatives to the KWM were introduced in an effort to avoid perceived deficiencies of the KWM. For example, the KWM leads to shock waves and acceleration waves, both of which imply that vehicles can change speed instantaneously. Also, the KWM cannot be used to model the oscillatory solutions (stop-and-go traffic) that are seen in unstable traffic flow.

1.2 Alternative Models in Traffic Flow

This subsection is a very high-level description of traffic-flow models other than the KWM. For the reasons indicated in the preceding paragraph, *higher-order continuum models* were proposed [Michalopoulos et al. 1984]. Higher-order continuum models incorporate speed adaptation after certain time delay by including additional acceleration component, which also introduces diffusion, and hence smoothes the shock waves appearing in the KWM. For example, introducing dampening and viscosity leads to the following modified equation for speed:

$$\frac{\partial v}{\partial t}+v\frac{\partial v}{\partial x}=\frac{1}{\tau}(V_e(k)-v)-c_o^2\frac{\frac{\partial k}{\partial x}}{k}+\frac{1}{k}\frac{\partial(\mu(k)\frac{\partial v}{\partial x})}{\partial x} \quad (3)$$

Here ,

$\frac{1}{k}\frac{\partial(\mu(k)\frac{\partial v}{\partial x})}{\partial x}$ refers to the viscosity term,

$\frac{1}{\tau}(V_e(k)-v)-c_o^2\frac{\frac{\partial k}{\partial x}}{k}$ refers to the momentum term.

Higher-order continuum models have some difficulties, such as evolution of unexpected negative flows. As a result many other classes of traffic models have been proposed. One such class of models is *microscopic models* [2003]. The defining distinction between continuum and microscopic models of traffic flow is that the latter retain the discrete characteristic of actual flows, through attempting to simulate the motion of individual vehicles. Suffice it to say here that these days they are largely favored over continuum models, especially in the U.S. The reasons for that might have some to do with doubts about the validity of continuum representation, but probably they have much more to do with other doubts about the KWM. The ability to provide simple and intuitively appealing graphical output from microscopic models also is undoubtedly a factor in the prevailing preference for microscopic models. However, the price paid for these features may include the necessity for a large number of independent runs to "average out" the statistical fluctuations necessarily present in any single simulation run with a microscopic model.

An example of microscopic models is car-following models [Kuhne et al. 2002]. These models are usually represented in discrete-time continuous-space form, although models represented as continuous-time continuous-space and discrete-time discrete-space has also been proposed. The car-following model, in all the cases, attempts to quantitatively represent the response of a driver to the stimulus of this speed and position relative to other vehicles ahead. For a typical discrete-time continuous-space model, in each time step we process the vehicles as follows: The vehicle farthest from downstream is moved at its current speed for the time step, if its speed is same as free-flow speed, else it is accelerated. All other vehicles are then processed, starting from the vehicle farthest downstream progressing to the vehicle farthest upstream. These vehicles are put under constraints such as: it cannot exceed the free flow speed; it should maintain safe position relative to car ahead; and it cannot exceed its performance capabilities.

Microscopic models deal with individual vehicles and their interaction but are less useful when applied to a network. A traffic network consists of network topology and a traffic control system which consists of traffic signals, lane configuration etc.

1.3 Thesis Layout

Chapter II deals with the method of Godunov, which is the primary discrete computational approximation employed in this thesis. This chapter will also provide an overview of Riemann problems for the KWM, because this same methodology will be employed subsequently to develop discrete computational approximations to mathematically continuous boundary, interface and point-constriction conditions.

Chapter III contains description of the mathematical boundary, interface and point-constriction conditions, and development of discrete counterparts of these that are suitable for implementation within the Godunov method. The development of these discrete approximations will be based upon exact solutions of suitable Riemann problems.

Chapter IV is dedicated to description of three useful scenarios, proposed by Ross. We discuss the scenarios in detail along with our computational solutions of these. These solutions are based upon MATLAB implementation of the developments of Chapter III. The primary features of these computational solutions are described in terms of the principal constructs of solutions of the KWM, notably shock waves and acceleration waves. The latter provides the quantitative, or at least semi-quantitative, verification of our discrete approximations that is one of the major objectives of this thesis.

Chapter V deals with extensions provided to the LWR simulation to make it generic and user friendly. This may include providing a graphical interface selecting appropriate simulation and specifying the important parameters.

Chapter VI summarizes the results obtained from the simulation and the inferences drawn. It also discusses briefly about the possible extensions possible to the simulation and means to carry out those extensions.

CHAPTER II

GODUNOV METHOD

Kinematic-wave model (KWM) is a first order traffic flow model which can be used to predict traffic flow. To solve this flow model numerically, many discretization techniques have been proposed. Godunov method [Lebacque 1996], which was originally proposed by Godunov to solve nonlinear hyperbolic equations, of which the KWM is an instance, is one such method.

2.1 Godunov Method

We shall start by describing, following Lebacque [1996], this method in its application to non linear hyperbolic conservation equation of the form:

$$\begin{aligned} &\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \\ &u(x,0) = u_o(x) \end{aligned} \qquad \forall x \in \Re \qquad (4)$$

Here,

u is an unknown function of a real variable x.

f(u) is an known function of u.

In the following chapter III, we discuss modification to account for finite boundary locations ($\forall x \in \Re \rightarrow x \in [a,b]$, for finite a and b) and piece-wise dependency of the function *f* on position ($f(u) \rightarrow f(u,x)$ where *f* is piece-wise constant in X).

Following is the description of the scheme for this context. In this the spatial distance is denoted by x and time is denoted by j.

Step 1: Discretize the x-axis into small cells (a) = [x_{a-1} , x_a] of length $l_a = x_a - x_{a-1}$.

Step 2: Discretize the time axis into small interval [$j\Delta j, (j+1)\Delta j$].

Step 3: At any time step, $j\Delta j$, the solution is approximated by a piecewise constant function $\widetilde{u}$. Although this function is assumed to be constant in the cell being observed, its value can vary from cell to cell. This is defined notationally as:

$$\widetilde{u}(x, t\Delta t) \overset{def}{=} u_a^t \quad (\forall x \in (a))$$

Step 4: To compute the solution at the next available time step, first calculate the exact solution of (4), given the initial conditions at $j\Delta j$. This problem can be represented as:

$$\frac{\partial \Xi}{\partial t} + \frac{\partial}{\partial x} f(\Xi) = 0$$
$$\Xi(x, j\Delta j) = \begin{cases} \widetilde{u}(x, j\Delta j) & (\forall x \in \Re) \\ u_a^j & (\forall (a), \forall x \in (a)) \end{cases} \tag{5}$$

This is a Riemann problem. Such problems are discussed in subsection 2.3 below.

Step 5: The solution for the next time step for a particular cell *(a)* can be found by taking the average over the cell under consideration.

$$u_a^{j+1} = \frac{1}{l_a} \int_{(a)} \Xi(y, (j+1)\Delta j) dy \tag{6}$$

Equation (4) and (5) can also be written as,

$$u_a^{j+1} = u_a^j + \frac{\Delta j}{l_a} (\Phi_{a-1}^j - \Phi_a^j) \tag{7}$$

$$\Phi_a^j \overset{def}{=} \frac{1}{\Delta j} \int_{j\Delta j}^{(j+1)\Delta j} f[\Xi(x_a, s)] ds \tag{8}$$

Here, Φ_a^j represents the average flow crossing the cell x_a at time j.

Also, if the function *f(u)* is concave in nature, (7) can be replaced by the following Osher formula:[Lebacque 1996]

$$\Phi_a^t = \begin{cases} Min_{u_a^j \le u \le u_{a+1}^j} f(u) & if\ u_a^j \le u_{a+1}^j \\ Min_{u_{a+1}^j \le u \le u_a^j} f(u) & if\ u_a^j \ge u_{a+1}^j \end{cases} \tag{9}$$

To find the solution, keep on executing steps 3, 4 and 5 in a loop. As we can determine the value of function *u(j+1)* at time step *t+1*, given the value of *u* at time step *j* and also that we know the initial conditions, we can find out the value at all the time steps for all the cells.

2.2 Godunov and KWM

Godunov method is used to find solution for hyperbolic equations. To be able to use it for KWM we need to put the governing equations of KWM in the same form as expected by Godunov method. The basic premises of KWM are the conservation of vehicles and the definition of flow as a function of vehicular density. The equations governing these premises for a section of homogenous lane without any entry or exit points except the boundary are given as,

$$\frac{\partial k}{\partial t} + \frac{\partial q}{\partial x} = 0 \tag{10}$$

$$q = Q(k(x,t),x,t), \tag{11}$$

We can rewrite Equation (10) as,

$$\frac{\partial k}{\partial t} + \frac{\partial Q(k(x,t),x,t)}{\partial x} = 0 \tag{12}$$

This equation is now of the form of (4), so we can apply the Godunov method to find out the solution. For this, consider a section of road which needs to be modeled using KWM. To find the solution, we now apply Godunov method.

The first step is to decide the length of the road section and the time step for observation.

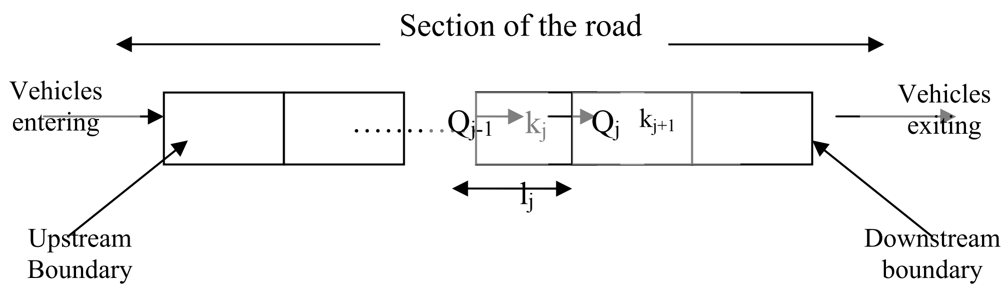


Figure 2: Schematic representation of the road section

The whole section of road (Figure 2) is broken down into smaller links. Each link is assumed to have a constant flow function. The time step and the link length are chosen in such a way that solutions computed at neighboring links do not interfere with each other. This is important because if this condition is not met, computations are still possible, but the results obtained such are found inconsistent with the actual data. This is shown in chapter 4, which would show the simulation results. This condition is also known as Courant, Friedrichs and Lewy (CFL) condition. So each link obeys the following condition [Lighthill, M. J., and Whitham, G.B. 1955]:

$$\Delta x / \Delta j \geq u_f$$

Here,

Δx is the length of the subsection.

Δj is the time slice of observation.

u_f is the free flow speed.

Applying steps 3, 4 and 5 of Godunov method leads us to the following equations [Michalopoulos et al. 1984]:

$$k_a^{j+1} = k_a^j + \frac{\Delta j}{l_a}(Q_{a-1}^j - Q_a^j) \tag{13}$$

$$Q_a^j \overset{def}{=} \frac{1}{\Delta j} \int_{j\Delta j}^{(j+1)\Delta j} \Phi(x_a, s) ds \tag{14}$$

If the flow function is concave in nature, we can apply the Osher expression to obtain the following equation:

$$Q_a^j = \begin{cases} Min_{k_a^j \le k \le k_{a+1}^j} Q_e(k) & if \quad k_a^j \le k_{a+1}^j \\ Min_{k_{a+1}^j \le k \le k_a^j} Q_e(k) & if \quad k_a^j \ge k_{a+1}^j \end{cases} \tag{15}$$

The derivation of Osher's formula and the clue to treatment of spatial discontinuity can both be addressed under the Riemann Problem which is the next topic of discussion.

2.3 Riemann Problem

The Riemann problem deals with solving a system in which the initial state is defined in terms of two constant states that are separated by a discontinuity (Fig. 3)

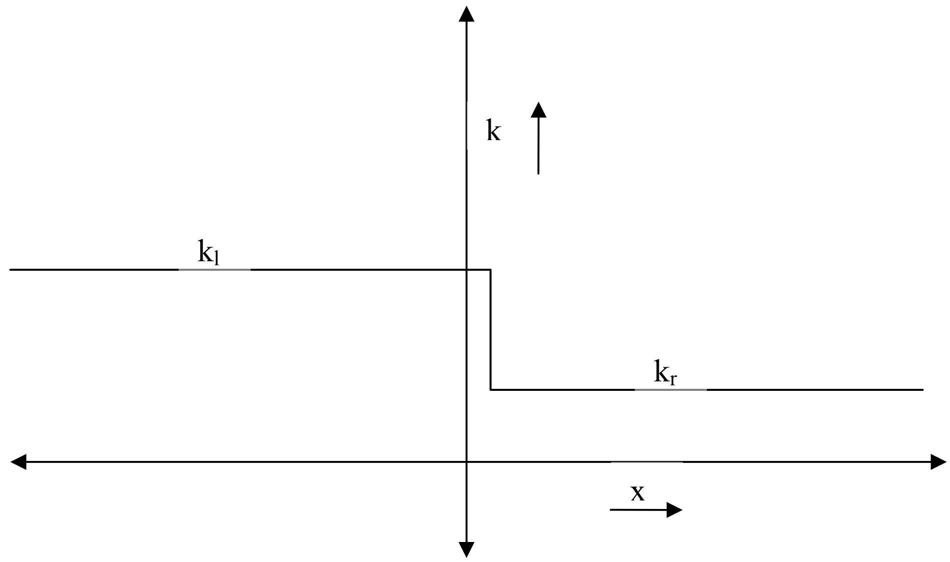


Figure 3: A typical initial condition of Riemann problem

In the present section, we will solve a generic Riemann problem. Consider a scenario where the initial density depends explicitly and discontinuously on the x-coordinate [Lebacque 1996]. This scenario is applicable in the cases where we are trying to model intersections (where there is a time dependent spatial discontinuity), variable number of lanes or traffic incidents (which can lead to temporary change in traffic parameters like capacity of the roadway, free flow speed etc.). To estimate the equilibrium flow Q_a^j (14) using Riemann method, we observe the following [Lebacque 1996]:

Observation 1: The initial condition for KWM is specified in terms of the vehicular density at the start of the observation point.

$$k(x,0) = \begin{Bmatrix} k_l & if & x < 0 \\ k_r & if & x > 0 \end{Bmatrix} \tag{16}$$

Here,

The subscript *l* and *r* refer to left and right of the point of observation.

Observation 2: The flow is piecewise constant. This means that in a cell, we take the equilibrium value (Q_e) for computation. The flow value changes only across the cell boundary which is assumed to be at the origin in the following equations:

$$Q_e(k,x) = \begin{Bmatrix} Q_e(k,l) & if & x < 0 \\ Q_e(k,r) & if & x > 0 \end{Bmatrix} \tag{17}$$

Observation 3: There exist characteristics lines of flow in the x-t plane. These lines have the property of constant flow along them. Due to the property of constant flow, these characteristics lines also have constant density along them. The lines are given by the following equations:

$$\begin{Bmatrix} x(t_0) = x_0 \\ \dfrac{dx}{dt} = \dfrac{\partial Q_e}{\partial k}[\ Q_{e*}^{-1}(Q_x, x), x] \end{Bmatrix} \tag{18}$$

Here, * refers to either free flow or congested flow. There are two densities for each flow value, one lies in the free flow region and the other lie in the congested region. The existence of multiple densities is result of the assumption that the flow function is concave in nature.

Observation 4: Density on a characteristic line is obtained from the flow value. This is done by using the equilibrium flow-density relationship corresponding to the traffic state. This also implies that the density is constant along a characteristic line.

Observation 5: The cell boundaries are chosen in such a way that the solution computed in one cell does not affect the other cell. This is done by choosing the cell length such that $\Delta x / \Delta t \geq u_f$. This condition is called as CFL condition.

Observation 6: Another important factor to be considered which choosing cell boundaries is making sure that all the spatial discontinuity points lie on one of the boundary. The spatial discontinuity points are those points where there is a discontinuity in flow value. This is important to ensure that we can safely consider the flow in cell to be piecewise constant.

2.3.1 Solutions of Riemann Problem

The following solutions are obtained after analyzing the characteristic curves for various combinations of densities across the spatial discontinuity. All the stated cases will be discussed in detail in chapter III.

2.3.1.1 Solutions for $Q_{max}(l) > Q_{max}(r)$

The solutions for this case is given in Table I. Here, Q_{max} refers to the maximum possible flow. It might happen that the maximum permissible flow left of the discontinuity is different from the maximum permissible flow to the right of discontinuity. A situation where this can happen is during a lane drop. For example, if the number of lanes drop from three to two and each lane is able to carry a maximum flow of Q_{lane}, then $Q_{max}(left) \geq Q_{max}(right)$.

Q_{left}, Q_{right} refer to the flow at the left and right of a boundary point.

k_l, k_r refers to the density at the left and right of a boundary point.

Undercritical refers to free flow.

Overcritical refers to congested flow.

Table I. Solution for incoming flow greater than outgoing flow

$k_l \setminus k_r$	Undercritical	Overcritical
Undercritical	Min [Q_{left}, Q_{max}(right)	Min [Q_{right}, Q_{left}]
Overcritical	Q_{max}(right)	Q_{right}

2.3.1.2 Solutions for $Q_{max}(l) < Q_{max}(r)$

The solutions for this case is given in Table II. Here, Q_{max} refers to the maximum possible flow. It might happen that the maximum permissible flow left of the discontinuity is different from the maximum permissible flow to the right of discontinuity. A situation where this can happen is during a lane expansion. For example, if the number of lanes increases from two to three and each lane is able to carry a maximum flow of Q_{lane}, then $Q_{max}(right) \geq Q_{max}(left)$.

Q_{left}, Q_{right} refer to the flow at the left and right of a boundary point.

k_l, k_r refers to the density at the left and right of a boundary point.

Undercritical refers to free flow.

Overcritical refers to congested flow.

Table II. Solution for incoming flow less than outgoing flow

$k_l \setminus k_r$	Undercritical	Overcritical
Undercritical	Q_{left}	Min [Q_{right}, Q_{left}]
Overcritical	Q_{max}(left)	Min [Q_{right}, Q_{max}(left)

CHAPTER III

BOUNDARY AND INTERFACE CONDITIONS

As shown in Chapter II, the Godunov method may be used to solve the hyperbolic equations that form the basis of KWM. One of the biggest challenges that arise in determining the solution is being able to identify correctly the boundary conditions at each of the links making up the system. This chapter looks at the principles of demand and supply and its consequent application in defining boundary conditions.

3.1 Demand and Supply

Consider a section of roadway divided into smaller links where each link is defined by a flow function $q = Q(k,x,t)$ and density k. We now try to define the quantities demand and supply for each link.

For any link, demand can be defined as the maximum traffic flow that can exit the link regardless of the condition downstream. This situation can be approximated as the downstream consisting of an infinite parking lot. This parking lot can absorb any amount of traffic and hence does not limit the flow exiting the link in any way. The flow in this link is now completely dependent on the density upstream and the flow function in the link.

For any link, supply can be defined as maximum flow that can enter the link regardless of the situation upstream. Consider the scenario when there is an endless supply of vehicles at the upstream entrance which can supply traffic to fulfill any downstream demand for vehicles. Hence, the flow in this link is now dependent only on the density downstream of the link and the flow function in the link.

3.2 Boundary Conditions

For any link the demand and supply can be defined in terms of the vehicular density of the link.

$$S(k) = \max \ (Q(k') : k < k' < k_{jam}) \tag{19}$$

$$D(k) = \max (Q(k') : 0 < k' < k) \tag{20}$$

Here,

S(k) is the supply at the link's upstream boundary.

D(k) is the demand at the link's downstream boundary.

Q(k) is the flow in the interior of the link.

k_{max} is the jam density.

We will try to examine the boundary conditions. During analysis we also use the fact that for a density greater than the jam density, the flow drops down to zero. In other words, supply and demand for all densities greater than the jam density are zero. The following discussion is based on [Nelson 2004].

3.3 Upstream Boundary Condition

The upstream boundary condition evaluates the flow at the upstream boundary. We assume that there is a given time dependent demand ($D_b(\tau)$) at the entrance which is assumed to be at $x=0$. To evaluate the upstream boundary the following assumptions are taken [Nelson 2004]:

1. The flow at the boundary is determined using the supply and demand values at the boundary.

   $$q(0,t) = min\ (D_b(t), S(k_{sec}, 0+, t)) \tag{21}$$

   Here,

   k_{sec} is the vehicular density in the link,

   0+ refers to the a point just inside the link near the upstream boundary,

2. The density is assumed to be constant i.e. $k(x,t) \equiv k = constant$ for all $x \geq 0$.
3. The flow $Q_0(k) = Q(k, 0+, t+)$ is defined as the flow just inside the upstream boundary.
4. The supply for the link is defined as $S_0(k) = S(k, 0+, t+)$

The upstream boundary analysis can then be divided into four situations [Nelson 2004].

3.3.1 Congestion-limited Supply

In this scenario, the link under consideration is already in a congested state. Congestion is caused when the vehicular density in the link exceeds the density at which maximum flow takes place. Hence, the density in this link is greater than k_{max}. In this case, the supply in link determines the flow while the demand at the upstream boundary has no effect on the flow into the link. Thus the flow immediately upstream of boundary is equal to the supply, $S_0(k_{sec})$ and congested flow holds for some positive distance inside the boundary. Hence, the governing equations are:

$$Q_o(k_{sec}) = S_0(k_{sec}) < D_b(t+)$$

$$k_{sec} > k_{max}$$

While the flow is given by:

$$q(0,t+) = Q_0(k_{sec})$$

3.3.2 Capacity-limited Supply

In this scenario, the link under consideration has supply equal to the capacity of link. Hence, it already has supply equal to maximum permissible flow, so that the demand D_b is greater than the flow. In this case, the flow at the boundary is limited by the capacity of the link. It no longer depends on the demand profile. This results in generation of an acceleration wave connecting a larger upstream density (k_{max}) to a lower downstream density region (k_{down}). In such case, the flow across the boundary is equal to the max flow, Q_{max}. This flow resembles a queue discharge or an acceleration wave. The governing equations are:

$$S_o(k_{sec}) = Q_{max}(0,t_j+) < D_b(t+)$$

$$k_{sec} \leq k_{max}(0+, t_j+)$$

While the flow is given by:

$$q(0,t+) = Q_{max}(0+,\ t+)$$

3.3.3 Demand Control (Queue Discharge)

In this scenario, the link under consideration has demand less than the supply and the sectional density less than the density at which the maximum flow occurs. Now, this also means that there is a value of density at which the flow is equal to the demand at the boundary. Because of the concave nature of the flow curve, there are actually two such densities. One density will be less than k_{max} while the other density will be greater than k_{max}. If this density is greater than the link density, an acceleration wave is generated connecting the upstream region of higher density to the downstream region of lower density. In such case, the flow across the boundary is equal to demand value D_b. This flow resembles a queue discharge or an acceleration wave. The governing equations are:

$$D_b \leq S_o(k_{sec})$$

$$k_{sec} \leq k_q < k_{max}$$

Here,

k_q is the density at which *flow* = D_b.

While the flow is given by:

$$q(0,t+) = D_b\,(t+)$$

3.3.4 Demand Control (Shock Wave)

This case is similar to the preceding one except for the fact that link density is greater than k_q. In this case a shock wave is generated connecting lower upstream density to higher density downstream. Even in this case, the flow at the boundary is given by the demand value D_b. The governing equations are:

$$D_b \leq S_o(k_{sec})$$

$$k_q \leq k_{sec} < k_{max}$$

While the flow is given by:

$$q(0,t+)= D_b\ (t+)$$

3.4 Downstream Boundary Condition

The downstream boundary condition evaluates the flow at the downstream boundary. Normally in analysis we assume that there is sink of infinite capacity at the downstream end, *at* $x=a$, but in the downstream boundary condition we assume that this is not the case and there is a time dependent downstream supply. To evaluate the boundary the following assumptions are taken [Nelson 2004]:

1. The flow at the boundary is determined using the supply and demand values at the boundary.

$$q(a,t) = \min\ (D(k_{sec}\ (a\text{-},\ t+),\ a\text{-},\ t),\ S_b(t)) \qquad (22)$$

2. The density is assumed to be constant i.e. $k(x,t) \equiv k_u = constant.$
3. The flow just inside the link, at the downstream boundary is, $Q_0(k) = Q(k,\ a\text{-},\ t+)$.
4. The demand for the link is defined as $D_0(k) = D(k,\ a\text{-},\ t+)$

The boundary analysis can then be divided into the following situations [Nelson 2004].

3.4.1 Demand Starvation

In this case, the demand at the boundary is less than the supply. Hence, there is always more room for more traffic than is being provided by the demand. Due to this, the flow at the boundary is controlled by the demand at the exit. This is free flowing traffic. The governing equations are:

$$D_a(k_u,\ a\text{-},\ t) = Q_a(k_u) \leq S_b(t)$$

$$k_u \leq k_{max}(a\text{-},\ t+)$$

While the flow is given by:

$$q(0,t+)= Q_a(k_u)$$

3.4.2 Capacity-limited Demand

In this case, the density in the link is more than the density at which maximum flow occurs. Also, the demand at boundary is equal to maximum allowable flow. Here the supply is greater than the demand, so the flow across the boundary becomes limited by the capacity of the link. This leads to generation of an acceleration wave which connects the higher upstream density of k_u to lower downstream density of k_{max}. The governing equations are:

$$D_a(k_{sec}, a-, t) = Q_{max}(a-, \tau+) \leq S_b(t)$$

$$k_{sec} \geq k_{max}(a-, t+)$$

While the flow is given by:

$$q(0,t+) = Q_a(k_u)$$

3.4.3 Supply Control (Queue Discharge)

In this case, the demand in the link under consideration is more than the supply. In addition the sectional density is less than the density at which the maximum flow occurs. Now, this also means that there is a value of density at which the flow is equal to the supply at the boundary. Because of the concave nature of the flow curve, there are actually two such densities. One density is less than the density at which maximum flow occurs (denoted by k_s), while the other is more than the density at which the maximum flow occurs. If the density in the link is greater than k_s, an acceleration wave is generated connecting the higher density upstream region (k_{sec}) to the lower density downstream region (k_s). This is free flowing traffic where the flow at the boundary is given by the supply value S_b. The governing equations are:

$$S_b \leq D_a(k_{sec})$$

$$k_s \leq k_{sec} < k_{max}$$

While the flow is given by:

$$q(0,t+) = S_b$$

3.4.4 Supply Control (Shock Wave)

This scenario is similar to the preceding one except for the fact that sectional density is less than k_s. In this case a shock wave is generated connecting upstream link of lower density (k_{sec}) to link of higher density downstream (k_s). This is free flowing traffic where the flow at the boundary is given by the supply value S_b. The governing equations are:

$$S_b \leq D_a(k_{sec})$$

$$k_{sec} \leq k_s < k_{max}$$

While the flow is given by:

$$q(0,t+) = S_b$$

3.5 Interface Conditions

Interface can be defined as a position where the TSM has a jump discontinuity [Nelson 2004]. Let the point of discontinuity be specified by ξ. The flow at the interface is defined in terms of demand and supply and the interface.

$$q(\xi, \tau) = min\ (\ D(k_u, \xi-, \tau+), S(k_d, \xi+, \ \tau+)\ ) \tag{23}$$

Here,

$q(\xi, \tau)$ is the flow at the interface.

$D(k_u, \xi-, \tau+)$ is the demand at the cell or link just before the interface.

$S(k_d, \xi+, \tau+)$ is the supply at the cell or link just after the interface.

k_u is the density just upstream of the interface.

k_d is the density just downstream of the interface.

$\tau+$ refers to time after which the solution at the interface is to be determined.

The interface scenarios can be divided into the following [Newell 1989]:

3.5.1 Congestion Limited Supply Control

This is the scenario when the density in the downstream link exceeds k_{max}. This case can now be analyzed as combination of an upstream boundary scenario of capacity-controlled supply control at upstream link of interface and downstream boundary scenario of congestion-limited supply control at the downstream link of the interface.

There are two different kind of wave generation depending on the upstream density. If the upstream density is less than k_d, then a shock wave is generated. If the upstream density is more than k_d, then an acceleration wave is generated. In both the cases, the flow across the interface is given by the supply. The governing equations are:

$$S_d(k_d) \leq D_u(k_u),$$
$$k_d \geq k_{max}$$

While the flow is given by:

$$q(0,t+) = Q(\xi) = S_d(k_d)$$

3.5.2 Capacity Limited Supply Control

In this scenario, the density just downstream (ξ+) is k_{max} while the rest of downstream density is less than k_{max}. Due to this the supply at the interface is same as Q_{max} while the demand at the interface is greater than the supply. In such a case, there is an acceleration wave originating in the downstream end. On the upstream end, there is generation of acceleration wave if the density upstream is less than k_{max} or a shock wave is generated otherwise. In such a case, the flow at the interface is given by Q_{max}. The governing equations are:

$$S_d(k_d) = Q_{max}(\xi+,\ \tau+)\ \leq D_u(k_u)$$
$$k_d \geq k_{max}(\xi+,\ \tau+)$$

While the flow is given by:

$$q(0,t+) = Q(\xi) = Q_{max}(\xi+,\ \tau+)$$

3.5.3 Capacity Limited Demand Control

This scenario is very similar to the one defined in the preceding subsection. In this the upstream density exceeds k_{max} and the demand upstream is equal to $Q_{max}(\xi-,\ \tau+)$. In this case, an acceleration wave is generated connecting the higher upstream density to ξ-density. On the downstream side, acceleration wave is generated if k_d is less than k_{max}, else a shock wave is generated. The governing equations are:

$$D_u(k_u) = Q_{max}(\xi+,\ \tau+)\ \leq S_d(k_d)$$
$$k_u \geq k_{max}(\xi-,\ \tau+)$$

While the flow is given by:

$$q(0,t+) = Q(\xi) = Q_{max}(\xi-, \tau+)$$

3.5.4 Demand Starvation

In this scenario, the demand at interface is same as upstream flow, while the supply at downstream is more than the demand. The other important condition is that $k_u \leq k_{max}(\xi-, \tau+)$. In this case, there is an shock/ acceleration wave generated at downstream end $(\xi+)$ depending on k_d >/< $k(\xi+)$. The governing equations are:

If,

$$D_u(k_u) = Q_u(\xi-, \tau+) \leq S_d(k_d)$$

$$k_u \leq k_{max}(\xi-, \tau+)$$

While the flow is given by:

$$q(0,t+) = Q(\xi) = Q_u(\xi-, \tau+)$$

3.6 Point Constrictions

Till now we have seen cases in which there was no explicit dependency on time. Such time dependent flow functions are seen in cases like accidents, signalized intersections or incidents [Newell 1989]. To model these cases, the incidents are modeled as happening in a single point in space and time. This is done by introducing a time-varying capacity component in determining the flow [Nelson 2004].

$$q(\xi, \tau) = min\ (\ D(k_u, \xi-, \tau+), S(k_d, \xi+, \tau+), C(\xi, \tau+)\ ) \qquad (24)$$

Here,

$C(\xi, \tau+)$ refers to a time-dependent function which denotes the capacity of roadway at the constriction point ξ.

Analysis of point constriction is similar to the analysis of interface scenarios. If the value of the capacity function is more than that of the supply or demand function, the function has no effect on the analysis. In other words, the minimum value at ξ determines the analysis.

If the minimum is given by the capacity function, then the analysis is done separately for ξ- and ξ+. For the upstream end, the capacity now functions as downstream supply and is analyzed as supply controlled cases in the preceding section. For the downstream end, the capacity functions as upstream demand and is analyzed as demand controlled case.

CHAPTER IV

ROSS SCENARIOS

In chapter III, we looked at the boundary and interface conditions. To prove the validity of these conditions, we are now going to apply them to solve a set of scenarios. These scenarios are also known as Ross scenarios as they were first seen in a paper by Paul Ross [1988]. Most of the simulation results discussed in this chapter is part of [Nelson 2004].

4.1 Ross Formulation

Before we start with Ross Scenarios, let us briefly look at the formulation used by Ross to obtain his results [Ross 1988]. Ross simply expressed the expression as:

$$\frac{\partial v}{\partial t} + v * \frac{\partial v}{\partial x} = \frac{(F - v)}{T} \text{ , } k \leq k_{jam} \tag{25}$$

Where,

v denotes the traffic speed,

k_{jam} denotes the jam speed,

F denotes the free flow speed of the traffic (a constant),

T denotes the relaxation time.

4.2 Introduction to Ross Scenarios

The Ross scenarios were simulated using a program written in MATLAB® and the obtained results were then compared with the expected result. The first step in setting up the simulation was defining the various parameters and an appropriate TSM. The following parameter values were used as defined in the Ross paper [1988].

Free-flow speed (v_{free})= 63 mph,

Capacity of the roadway (c_{max}) = 2000 vphpl,

Jam density (k_{jam})= 143 vpmpl.

The next step was deciding on a TSM. The TSM used through out during the simulation was the triangular TSM, as defined by Newell [1989]. The TSM defined for a single lane is as follows:

$$Q_1(k)=\begin{cases} v_{free} * k\, vph, & 0 \le k \le c_{\max}/v_{free} \\ \dfrac{c_{\max}}{(k_{jam} - c_{\max}/v_{free})} * (k_{jam} - k) & c_{\max}/v_{free} \le k \le k_{jam} \end{cases} \tag{26}$$

The above TSM is defined for one lane. We define the aggregate TSM as:

$$Q_n(k_{agg}) = n * Q_1(k_{agg}/n)$$

Here,

Q_n is the flow function for multilane system consisting of n lanes.

n represents the number of lanes.

k_{agg} represents the density aggregated over all the lanes.

This assumes that the lanes operate independent of each other and there is no interaction between them. Also, the vehicular traffic is assumed to be uniformly distributed across lanes.

In defining the conditions for Ross scenarios, the following notations are used:

a specified the length of the roadway under simulation.

T denotes the duration of simulation.

D_b is the demand at the entrant boundary.

S_b is the supply at the exit boundary.

C denoted the capacity of the roadway at any instant.

n represents the number of lanes.

We now go through the individual Ross scenarios.

4.3 Scenario 0 (Boundary Condition)

Before we start with actual Ross scenarios, it is worth while to discuss a very simple scenario dealing with the boundary conditions. The scenario is described by the parameters:

a = 13 miles,

T = 0.5 hour,

n = 3 lanes for $0 \leq x \leq 5$,

2 lanes for $5 \leq x \leq 13$,

D_b = 3000 vph,

S_b = 4000 vph,

k_{init} = 1500/63,

Simulation Parameters: $\Delta x = 0.1$ *miles,* $\Delta t = 0.001$ *hrs*

We shall ordinarily describe scenarios via a list of parameters, as above. The mathematically inclined may translate such a description into a mathematical problem, per the following example. The mathematical version of Scenario 0 is to use Godunov method, with $\Delta x=0.1$ miles and $\Delta t=0.001$ hours, to obtain a computational approximation to the (entropy) solution of the partial differential equation (12), for $0 \leq x \leq 13$ miles and $0 \leq \tau \leq 0.5$ hours, with flow function Q given by $Q_3(k) = 3 * Q_1(k/3)$ for $0 \leq x \leq 5$ miles and $Q_2(k) = 2 * Q_1(k/2)$ for $5 \leq x \leq 13$ miles, where Q is defined as in (26). Further, the initial condition is $k(x,0) = 1500/63$ vehicle per mile, the entrant boundary condition at $x = 0$ is (21) with $D_b(\tau) \equiv 3000$ vph, the exit boundary condition at $x = a = 13$ *miles* is (22) with $S_b(\tau) \equiv 4000$ vph, and the interface condition of subsection (3.5) is to hold at $x = 5$ miles.

This scenario can be approximated in real life as situation resulting from closing down of one lane in a two lane roadway coupled with sudden increase in entrant traffic (e.g. onset of rush hour). Let us now try to analyze the situation.

Initially there is constant traffic in roadway (specified by k_{init}). At the start of observation the entrant demand is D_b = 3000 and the entrant supply is S_b = 4000 vph. Hence this creates a demand controlled flow (section 3.3.3). This leads to generation of an acceleration wave connection the upstream density of k_u = 3000/63 to a downstream density of k_{init}. This acceleration wave is clearly seen in Figure 4 as increase in density at the origin and traveling towards the exit with a speed of $v = (3000 - 1500)/(k_u - k_{init}) = 63\ mph$.

The acceleration wave reached the five mile boundary after $\tau_1 = 5/63 \approx 0.794\ hrs \approx 4.8\ minutes$. When this wave reaches the boundary, it is met with a lane drop from two lanes to one lane. Hence, the supply at this interface is now defined by the capacity of the downstream lane while the demand is defined by D_b = 3000. This creates a capacity limited supply controlled flow (section 3.5.2) with *supply* = *capacity* = *2000* and *demand* = *3000*. On the upstream side, a shock wave is created connecting the upstream density of k_u to the downstream density of $k(\tau\text{-}) = k_{jam} + (2000/63) \approx 174.7\ mph$. This shockwave generation can be seen in Figure 4 as a sudden increase in density at the five mile boundary. This shockwave travels towards the origin with a speed of $v_{shock} = (2000 - 3000)/\ (k(\tau\text{-}) - k_u) = 1000/(174.7 - 47.6) \approx 7.87\ mph$. Hence, this shock wave is expected to reach origin in approximately $\tau_2 = 5\ /\ v_{shock} \approx$ 0.635 hrs ≈ 38.1 minutes. This shock wave is represented as the left edge facing the reader in Figure 4. As we can see that this reaches the origin in ≈ 0.73 hrs which is nearly equal to the predicted value of 42.9 minutes ($\tau_1 + \tau_2$). Once this shock wave reached the origin, it settles into steady state flow with $k_{upstream} = k_{jam} + (2000/63) \approx 174.7\ mph$ with flow $Q_{upstream} = 2000\ vph$. This is also very evident from the Figure 4, where this is shown in the plateau corresponding to density = 174.7 vpm in the *x-t* plane.

On the downstream side, an acceleration wave is generated connection the upstream density of $k(\tau_+) = 2000/63 = 31.74\ vpm$ to the downstream density of k_{init}. This acceleration wave travels with a speed of $v_{accel} = (2000 - 1000)/(31.74 - 23.8) = v_{free}$.

Once this acceleration wave reaches the exit boundary, the downstream link (link beyond the five mile boundary) settle down into steady flow of $Q_{downstream} = 2000\ vph$. This is also evident in the Figure 4 where the acceleration wave is clearly seen as a ripple originating at the five mile boundary and traveling till the exit boundary.

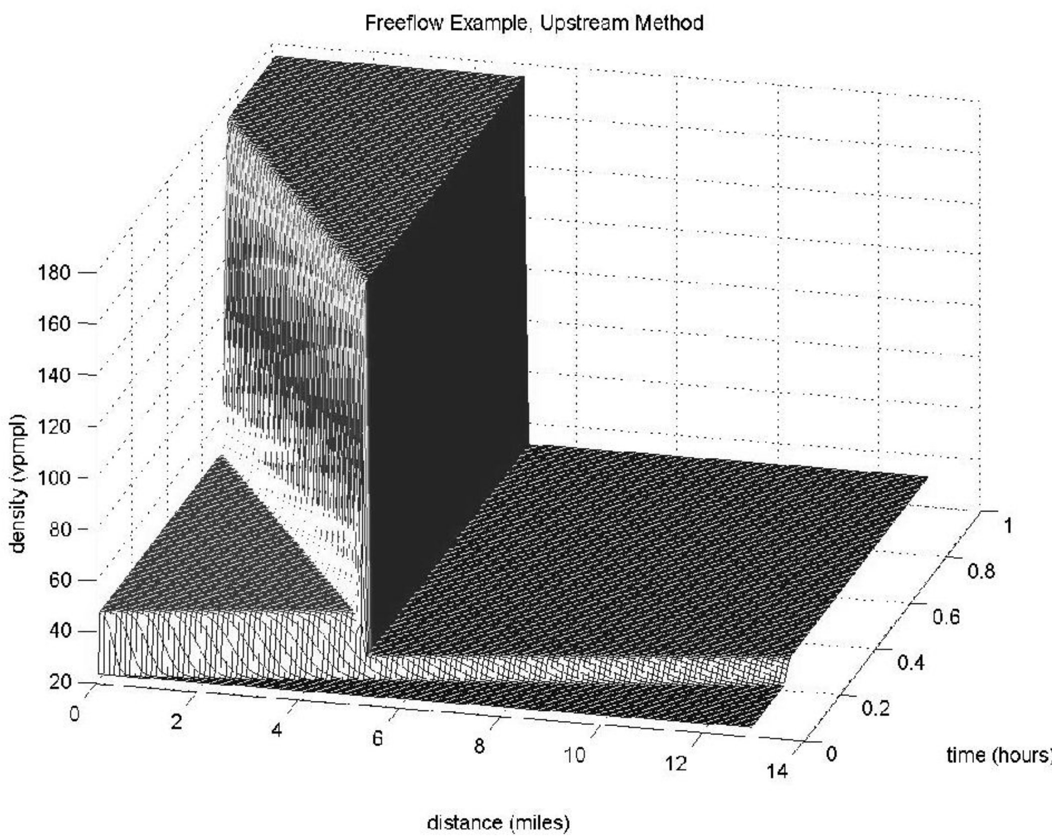


Figure 4. Simulation result for scenario 0 (boundary conditions)

4.4 Ross Scenario 2 (Interface Conditions)

Ross scenario 2 is defined as following [Ross 1988]:

a = 13 miles,

T = 2 hours,

n = 3 lanes for $0 \le x \le 5$,

2 lanes for $5 \le x \le 9$,

1 lanes for $9 \le x \le 13$,

$D_b(t)$ = 1800 vph for $0 \le t \le 0.1$,

2300 vph for $0.1 \le t \le 0.2$,

2800 vph for $0.2 \le t \le 0.3$,

3300 vph for $0.3 \le t \le 0.5$,

5800 vph for $0.5 \le t \le 1$,

S_b = 2000 vph,

Simulation Parameters: $\Delta x = 0.1$ *miles* , $\Delta t = 0.001$ *hrs*

Initial Condition 1: $k_{init} = 800/v_{free}$

The mathematical version of Scenario 2 is to use Godunov method, with $\Delta x=0.1$ miles and $\Delta t=0.001$ hours, to obtain a computational approximation to the (entropy) solution of the partial differential equation (12), for $0 \le x \le 13$ miles and $0 \le \tau \le 2$ hours, with flow function Q given by $Q_3(k) = 3^* Q_1(k/3)$ for $0 \le x \le 5$ miles and $Q_2(k) = 2^* Q_1(k/2)$ for $5 \le x \le 9$ miles and $Q_1(k)$ for $9 \le x \le 13$ miles, where Q is defined as in (26). Further, the initial condition is $k(x,0) = 800/63$ vehicle per mile, the entrant boundary condition at $x = 0$ is (21) with a time varying demand $D_b(\tau) \equiv 1800$ vph for $0 \le t \le 0.1$ and $D_b(\tau) \equiv 2300$ vph for $0.1 \le t \le 0.2$ and $D_b(\tau) \equiv 2800$ vph for $0.2 \le t \le 0.3$ and $D_b(\tau) \equiv 3300$ vph for $0.3 \le t \le 0.5$ and $D_b(\tau) \equiv 5800$ vph for $0.5 \le t \le 1$, the exit boundary condition at $x = a = 13$ *miles* is (22) with $S_b(\tau) \equiv 2000$ vph, and the interface condition of subsection (3.5) is to hold at $x = 5$ miles and $x = 9$ miles.

Ross [1988] describes the results for this scenario as follows: "A traffic queue forms upstream of the bottleneck (at 9.1 miles). In that queue the speed is 7.0 mph. The queue spills back, eventually filling the entire middle section of roadway. After the demand volume exceeds 4,000 veh/hr, another queue begins to form upstream from the bottleneck (at 5 miles). All phenomena seem representative of real traffic." [Ross 1988]

This scenario can be compared to a scenario where there is reduction in number of lanes from three to two at five miles and then finally to one after nine miles. This is also coupled with increase in the demand at the entrant boundary condition with time.

Let us try to analyze this scenario as combination of increase in demand and decrease in the number of lanes. The actual simulation result is shown in Figure 5 and Figure 6 while the result reported by Ross is shown in Figure 7. Initially the density along the roadway is k_{init} = 800/63 = 12.7 vpm. At the start of simulation the demand is increased to 1800 vph. There is an acceleration wave generated at the upstream boundary, connecting the upstream density of k_{u1} = 1800/63 to the downstream density of k_{init}. This acceleration wave propagates downstream with free flow speed and hence reaches the exit boundary in $\tau_1 = 13/63 \approx 0.2\ hrs$. The demand at the upstream boundary is still less than the capacity of the roadway section with one lane; hence this essentially means the only effect see due to this acceleration wave is increasing the vehicular density on the roadway. This effect is clearly seen in Figure 5, where the acceleration wave is the first ripple originating at the origin and reaching the exit boundary at approximately 0.2 hours.

The next increase in demand paints a very different picture. The demand at $t=0.1$ increases from 1800 vph to 2300 vph. This leads to creation of another acceleration wave connecting the upstream density of k_{u2} = 2300/63 =36.5 to the downstream density of k_{u1}. This wave propagates with the free flow velocity till it reaches the 9 mile interface. At this interface, the number of lanes is reduced to one with capacity of 2000 vpm. This leads to a capacity controlled, supply limited (section 3.3.2) condition at the interface. Hence, on the upstream side, a shockwave is generated which connects upstream density of k_{u2} = 2300/63 to the interface density of $k_{\tau 1-} = 143 + 2000/63 = 174.7\ vpm$. This shock wave propagates upstream with a velocity of $v_{wave1} = (2300 - 2000)/(36.5 - 174.7) = -2.2\ mph$. On the downstream side, there is an acceleration wave generated which connects upstream density of $k_{\tau 1+} = 2000/63$ to the downstream density of $k_d = 1800/63$. This effect is seen in Figure 5. The acceleration wave is seen as a ripple generated at $(x, t) = (0, 0.1)$ and propagating till the nine mile boundary, where we see the generation of shock wave which is seen as the curve. Ideally, this should have been a straight line instead of curve, but as we shall see later on, there are multiple shock waves generated and hence the curvature.

The next increase in demand is very similar because although the increase is from 2300 vph to 2800 vph, still there is no change in terms of capacities at the interface. Hence, the only change is terms of the wave propagating speed which now changes to $v_{wave2} = (2800 – 2000)/(44.4 – 174.7) = - 6.14\ mph$. The effect of increase in demand is seen in Figure 5. The acceleration wave is seen as a ripple generated at $(x, t) = (0, 0.2)$ and propagating till the nine mile boundary, where we see the generation of shock wave which is seen as the curve.

Demand is now increased to 3300 vpm, which is still not enough to cause shock wave at the five mile boundary. Hence, this is also treated the same way as the last two increments. The acceleration wave reflected from the nine mile boundary now moves at speed of $v_{wave3} = (3300 – 2000)/(52.4 – 174.7) = - 10.6\ mph$. This effect of increase in demand is visible in Figure 5. The acceleration wave is seen as a ripple generated at $(x, t) = (0, 0.3)$ and propagating till the nine mile boundary, where we see the generation of shock wave which is seen as the curve.

Demand is now increased to 5800 vpm. This increase would now definitely generate shock wave at the five mile boundary as it exceeds the capacity of the two lane roadway. Also we have seen the emergence of three shock waves from the nine mile boundary corresponding to *2300, 2800 and 3300 vph* demand at entrant boundary. These shock waves would affect the five mile boundary when they reach it. It would be interesting to see which wave, the acceleration wave due to increase in demand to 5800 vph or shock waves generated from the nine mile boundary, reach the five mile boundary first. This is important because this would determine the conditions and the nature of the waves generated at five mile boundary. We now try to determine the times at which these waves hit the five mile boundary. It is calculated as follows:

Time when the wave reaches the five mile boundary (t) = time when the wave was generated + time it takes to travel to the five mile boundary.

Time taken by acceleration wave generated due to increase in demand to 5800 vpl:

$$t_1 = 0.5 + 5/63 \approx 0.58\ hrs$$

Time taken by shock wave generated due to increase in demand to 2300 vpl:

$$t_2\ (for\ v_{wave1}) = 0.1 + 9/63 + 4/\ v_{wave1} = 2.06\ hrs$$

Time taken by shock wave generated due to increase in demand to 2800 vpl:

$$t_3\ (for\ v_{wave2}) = 0.2 + 9/63 + 4/\ v_{wave2} = 0.99\ hrs$$

Time taken by shock wave generated due to increase in demand to 3300 vpl:

$$t_2\ (for\ v_{wave3}) = 0.3 + 9/63 + 4/\ v_{wave3} = 0.83\ hrs$$

So, we see that the acceleration wave generated at the entrant boundary reaches the five mile boundary before any of the reflected shock waves reaches the five mile boundary. At this boundary, the number of lanes reduces from three to two, reducing the capacity to 4000 vpm. This now becomes a capacity limited, supply controlled situation (section 3.3.2). When the acceleration wave reaches the five mile boundary boundary, a shock wave is generated at the interface. The shock wave generated connects the upstream density of $k_u = 5800/63$ to interface density of $k_{\tau 2-} = k_{jam} + 4000/63 = 206.5\ vpm$. This wave then moves upstream with a speed of $v = (5800 - 4000)/(206.5 - 92.01) = 15.7\ mph$. On the downstream side, there is generation of an acceleration wave connecting $k_{\tau 2+} = 4000/63$ to downstream density of 3300/63. This shock wave is clearly seen in Figure 5 as white streak originating at five mile boundary and propagating towards the entrance boundary.

We now try to analyze the effect of the shock waves that were reflected from the nine mile boundary. The first of these waves reach the five mile boundary at approximately 0.83 hours. When the shock wave hits the boundary, the interface density changes from

4000/63 to $k_{\tau 1-} = 174.7\ vpm$. This changes the state from a capacity-limited supply controlled (section 3.3.2) to congestion-controlled supply controlled (section 3.3.1). This leads to generation of a shock wave which connects the upstream density of $k_{\tau 2-}$ to the new downstream density. The new downstream density is determined by finding out the maximum capacity flow that can happen on the three lane section. The maximum capacity is achieved at $k_{\tau 3-} = 2^* k_{jam} + Capacity\ flow\ on\ the\ third\ lane = 2^*143 + 2000/63 = 317.7\ vpm$. The shock wave thus generated travels upstream with a speed of $v_{wave4} = (5800 - 4000)/\ (317.7 - 206.5) = 16.2\ mph$. This shock wave is moving faster than the earlier generated shock wave and will finally overtake it. At that point the only shockwave remaining connects the upstream density of *5800/63* to downstream density of *317.7* and moving with a speed of $v_{wave5} = (5800 - 2000)/\ (317.7 - 92.1) = 16.8\ mph$. These features are clearly seen in Figure 5. The shock wave generated is seen as the slightly darker area above the white shock wave discussed in the preceding paragraph. The change in density is also very clearly seen in Figure 6 which shows the density change at section just before the five mile boundary. Two distinct spikes in density are seen. First at time $\tau = 0.58\ hours$ and the second at $\tau = 0.63\ hours$. The first spike is result of the acceleration wave reaching the five mile boundary while the second spike is result of the shock waves reflected from the nine mile boundary reaching five mile boundary.

We now compare the generated result with the one presented by Ross (Figure 7). The computational results obtained by simulation is very similar in nature to the one proposed by Ross, thus validating the simulation. We see the same kind of queue formations in Ross results as we see in the simulation result. But, there are a few places where they differ. The speed downstream of the nine mile boundary is predicted to be ≈ 7 mph in Ross, while from simulation it is ≈ 2000/174.7 ≈ 11.4 mph. As these are practically not very different, the difference is not of much consequence to us. Another difference is seen in the shock wave reflected from the 5 mile boundary. In the simulation, the shock gets back to the entrance approximately $t_1 + 5/15.7 = 0.9\ hrs$. On the other hand, the shock seems to be far from the entrance even at the end of one hour. Also, Ross does not say anything about the effect of the nine mile interface reflected waves in the density in the 5 mile section, which is a bit strange.

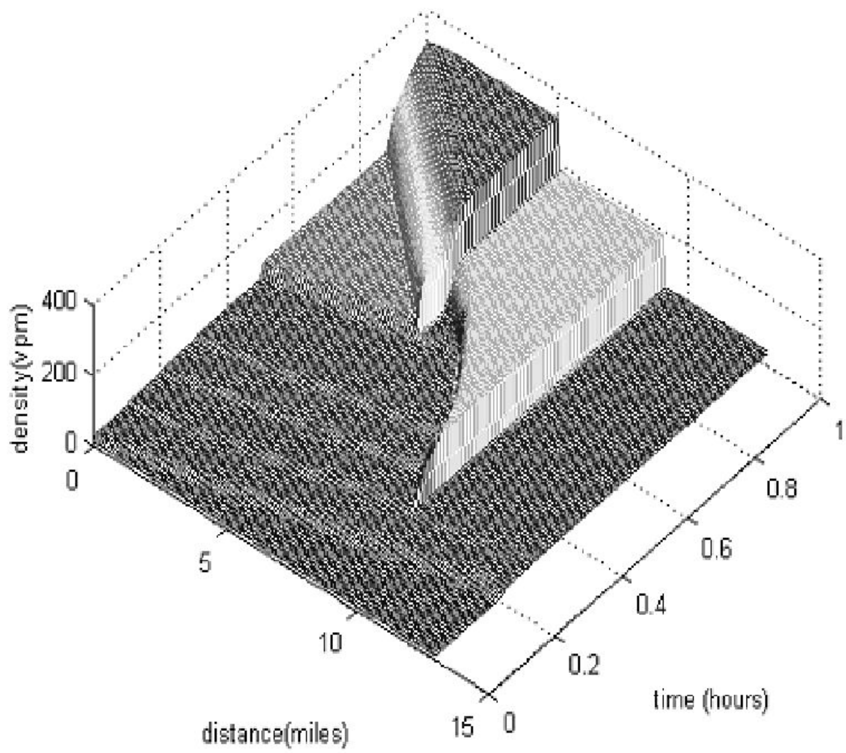


Figure 5. Simulation result for scenario 2 (interface conditions)

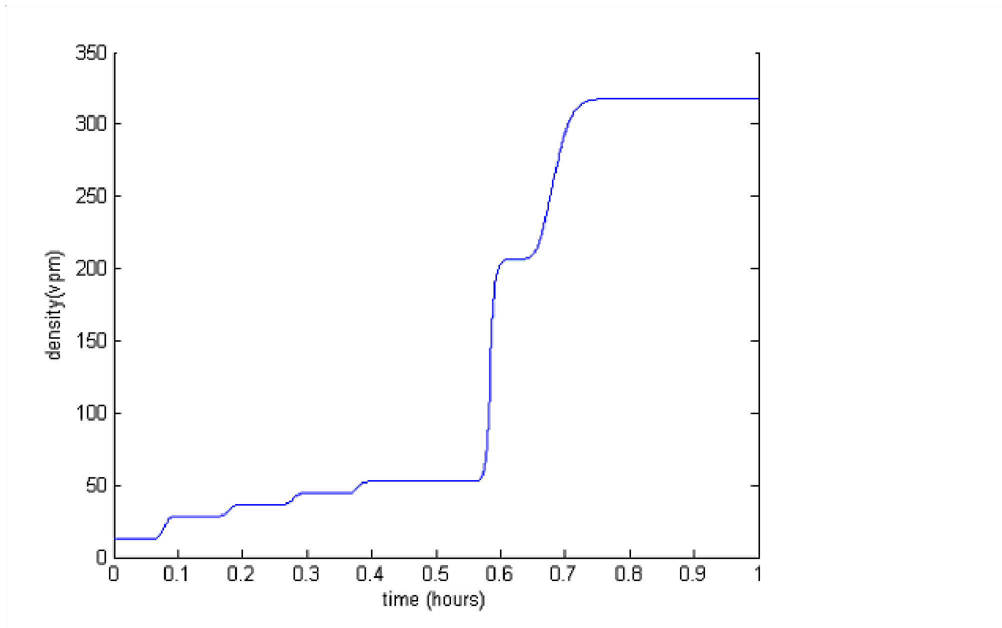


Figure 6. Density vs time plot at link just downstream of 5 mile boundary.

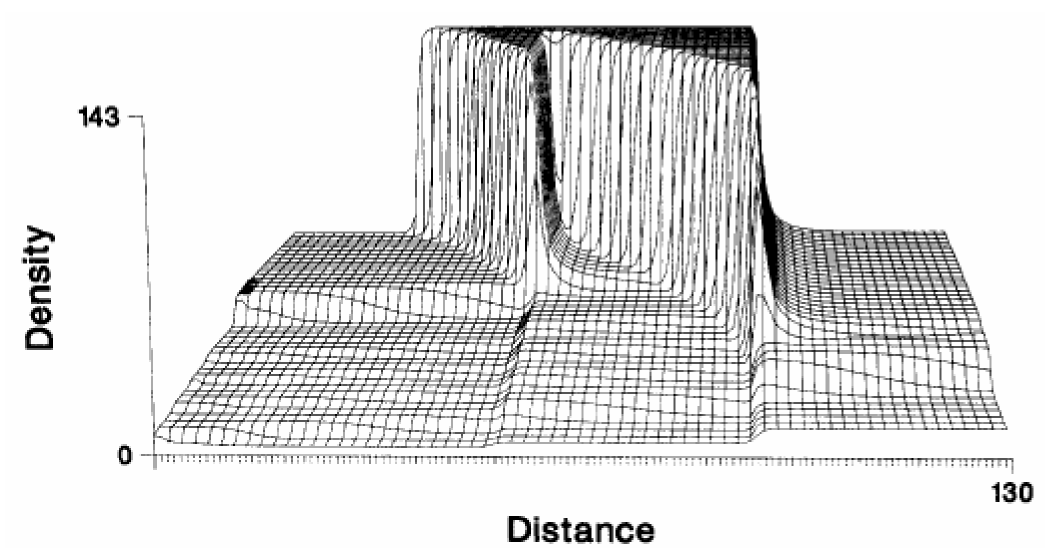


Figure 7. Result reported by Ross for scenario 2. Reprinted from Transportation Research B, Vol 22, Paul Ross, Traffic Dynamics, pp 421-435, Copyright (1988), with permission from Elsevier

4.5 Ross Scenario 1 (Static Point Constriction)

Ross scenario 1 is defined as following [1988]:

a = 13 miles,

T = 1 hour,

n = 2 lanes,

D_b *= 3000 vph,*

S_b *= 4000 vph,*

k_{init} *= 3000/63 = 47.6 vpm*

Special Condition 1: *C* reduces to 2000 vph at $x = 5.0$ miles for $0.2 \leq t \leq 0.5$ hrs

Special Condition 2: v_{free} reduces to 50 mph between $4.9 \leq x \leq 5.3$ for $0.2 \leq t \leq 0.7$

Simulation Parameters: $\Delta x = 0.1$ *mile,* $\Delta t = 0.001$ *hr*

The mathematical version of Scenario 1 is to use Godunov method, with $\Delta x=0.1$ miles and $\Delta t=0.001$ hours, to obtain a computational approximation to the (entropy) solution of the partial differential equation (12), for $0 \leq x \leq 13$ miles and $0 \leq \tau \leq 1$ hour, with flow function Q given by $Q_2(k) = 2* Q_1(k/2)$ for $0 \leq x \leq 13$ miles, where Q is defined as in (26). Further, the initial condition is $k(x,0) = 3000/63$ vehicle per mile, the entrant boundary condition at $x = 0$ is (21) with $D_b(\tau) \equiv 3000$ vph for $0 \leq t \leq 1$ hour, the exit boundary condition at $x = a = 13$ *miles* is (22) with $S_b(\tau) \equiv 4000$ vph, and the point constriction condition of subsection (3.6) is to hold at $x = 5$ miles for $0.2 \leq t \leq 0.5$ hrs with the free flow speed reduced to $v_{free} = 50$ mph between $4.9 \leq x \leq 5.3$ miles and $0.2 \leq t \leq 0.7$ hours.

Ross [1988] describes the results for this scenario, as obtained by computational solution of his proposed new model, as follows: “A traffic queue grows back from the accident site until the accident clears. (The queue appears as a region of high density and low speed.) The queue dissipates with volume = restored capacity but does not clear completely until about 0.33 hours after capacity returns to normal (about 0.13 hours after free-flow speed returns to normal). The volume downstream of the accident is decreased until the accident clears, whereupon it increases to the full capacity of the roadway."

In our simulation, we consider the original Ross scenario and a modified version of it. The difference between the two versions is the free flow speed. In the original Ross scenario there is a reduction in the v_{free} to 50 mph at the scene of accident. In the modified scenario, the free flow speed remains constant at $v_{free} = 63$ mph.

Let us try to analyze the scenario 1 [Nelson 2004]. The density along roadway continues to be k_{init} until the time of an event. At this time ($t = 0.2\ hrs$), the capacity of the roadway at the incident location changes, presumably because the number of available lanes is reduced. This reduction takes places at $x = 5\ miles$. We now apply the analysis of point constriction (section 3.6) to this. The reduction in capacity is to 2000 vph which is less than that of the $min\ (D_b, S_b)$, hence this creates shock waves. On the upstream side of the blockage, there is generation of shock wave which connects the upstream density of k_{init} to the downstream density of $k(\tau\text{-}) = k_{jam} + (2000/63) \approx 174.7\ mph$. So, this shock wave is actually traveling with velocity $= \Delta Q/\Delta k = (2000 - 3000) / (174.7 - 47.6) \approx -7.9\ mph$. Here the negative sign indicates that the wave is actually traveling upstream. On the downstream side of the blockage, there is a shock wave generation which connects the density $k(\tau+) = (2000/63) \approx 31.7$ to k_{init}. The shock wave thus generated moves downstream with a velocity $= \Delta Q/\Delta k = (3000 - 2000) / (47.6 - 31.7) \approx 63\ mph = v_{free}$. These two shock waves are visible in Figure 8 and Figure 9. The shock wave traveling upstream appears in Figure 9 as the edge facing the reader, while the shock wave traveling downstream appears in Figure 8 as the ripple at the back of figure.

Let us now consider the scenario after the blockage is removed. In this case at the interface the upstream demand and downstream supply both are 4000 vph. This is the case of capacity limited interface condition (section 3.5.2 and section 3.5.3) and hence the flow in both cases is 4000 vph. This leads to generation of a acceleration wave connecting the upstream density of *174.7 vpm* to the interface density $k(\tau-) = (4000/63) \approx 63.5$ *vpm*. The wave thus generated travels with velocity = $\Delta Q/\Delta k = (2000 - 4000) / (174.7 - 63.5) \approx -17.9$ *mph*. On the downstream side of the blockage, there is an acceleration wave generated which connects the upstream density of *174.7 vpm* to downstream density $k(\tau+) = (4000/63) \approx 63.5$ *vpm*. The acceleration wave moves downstream with the free flow speed, $v_{free} = 63$ *mph*. These waves are visible in Figure 8 and Figure 9. The acceleration wave traveling upstream appears in Figure 8 as the left front edge of the plateau, while the acceleration wave traveling downstream appears in Figure 9 as the ripple at the back of figure.

Let us analyze the interaction of various waves. The shock wave ($wave_1$) generated $t = 0.2$ *hrs* is traveling upstream with a velocity of *7.9 mph*. The acceleration wave ($wave_2$) generated at $t = 0.5$ *hr* is also traveling upstream with velocity of *17.9 mph.* Hence, When the blockage is cleared, $wave_1$ has already covered $x = 7.9 * 0.3 = 2.4$ *miles*. At this time the $wave_2$ starts. It will catch up with the $wave_1$ in $t = 2.4/(17.9 - 7.9) = 0.24$ *hrs*. At this time both waves merge into a single wave which connects the upstream density of *47.6* vpm to downstream density of *63.5 vpm*. This wave now travels downstream with velocity, v_{max}. On the downstream side, both the acceleration waves are propagating at the same speed. So, they do not overtake each other. All these waves are clearly seen in Figure 8.

The same interaction is seen in Figure 9, except for the higher density near the blockage. This can be compared in real life to the effect of “rubbernecking”. This is a commonly seen phenomenon where driver looks back to see the accident site while passing it. These lead to reduction in speed and hence increase in vehicular density near the accident site.

When we compare the results with the results reported by Ross (Figure 10), we find that there are two very obvious dissimilarities. The first one deals with queue dissipation. In

Ross's result the dissipation of queue appears to be happening from the upstream end while in our result, the queue dissipation appears to be happening from the downstream end. In real life, we normally see that the queue dissipates from the downstream end; hence we feel that our results are more representative of the real life scenario. The other difference is the distance which queue reaches upstream. In our simulation the upstream queue distance is much more than that of Ross's results. The possible reason could be that in Ross result, the queue starts dissipating from upstream end when the blockage is cleared, as opposed to simulation result where the queue continues to expand in that direction until overtaken by the later and faster acceleration wave generated by clearing the blockage.

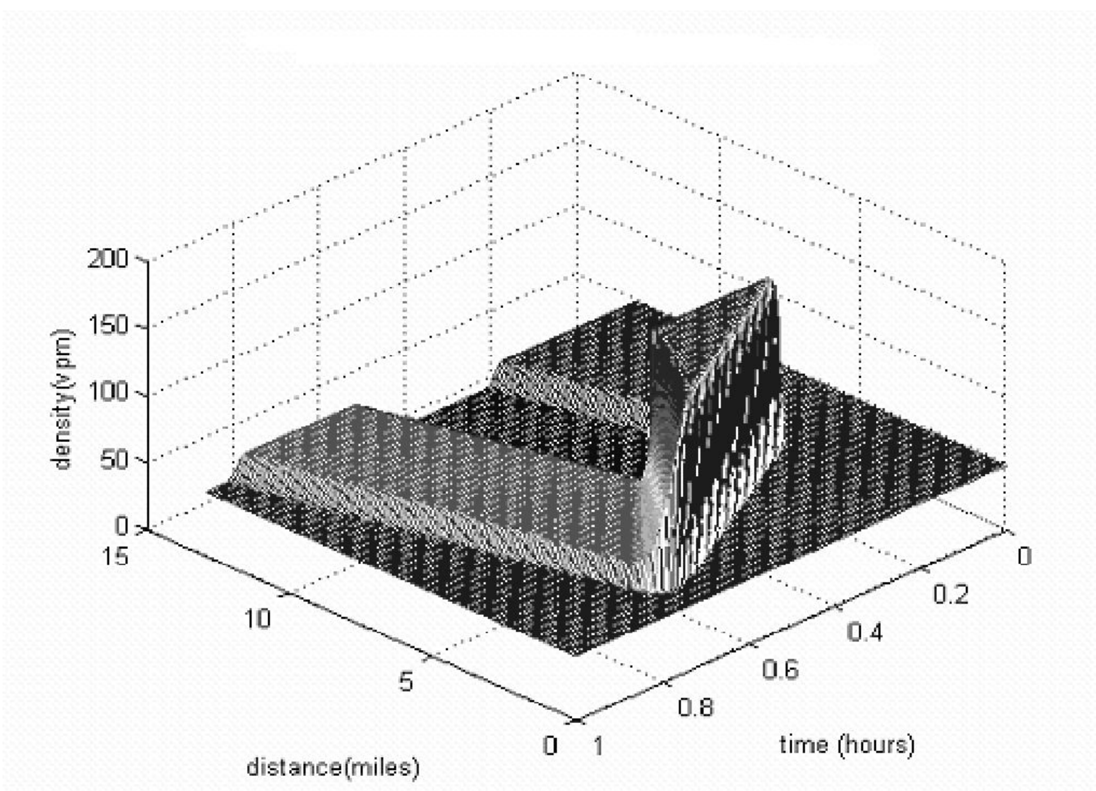


Figure 8. Simulation results for modified Ross scenario 1 (with constant v_{free})

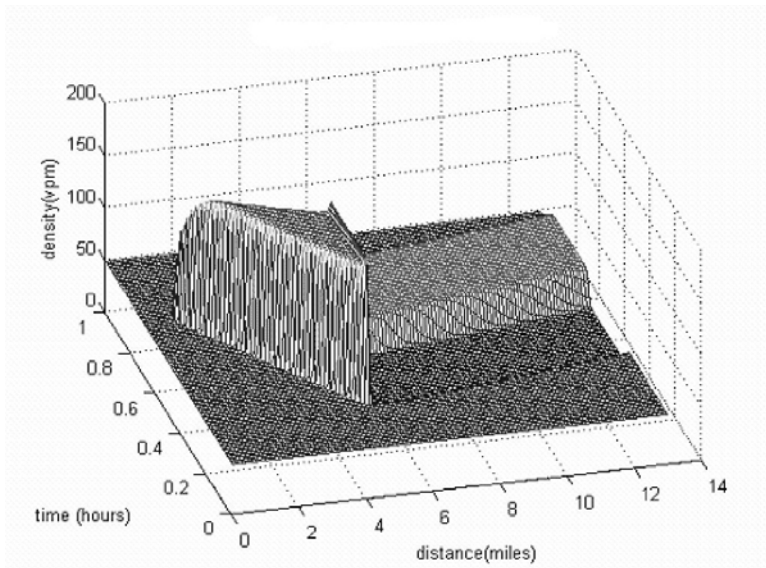


Figure 9. Simulation results for original Ross scenario 1

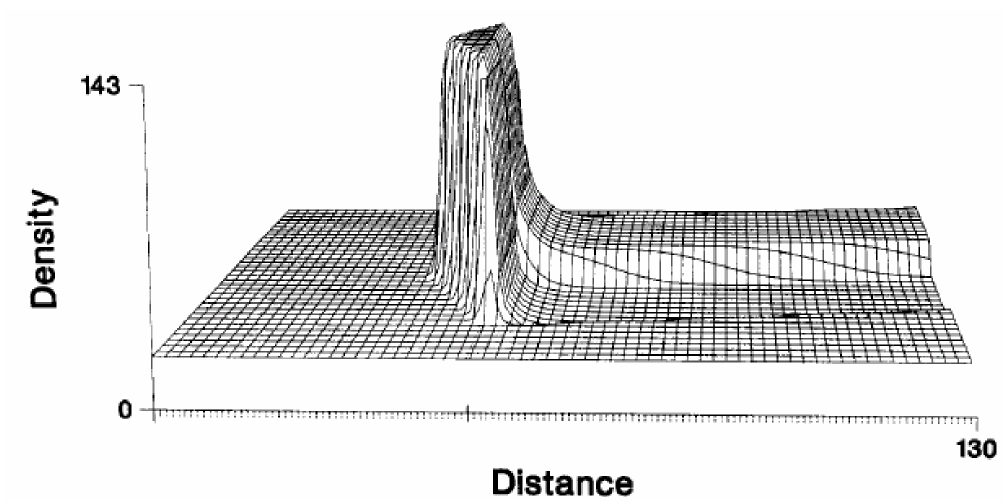


Figure 10. Result reported by Ross for scenario 1. Reprinted from Transportation Research B, Vol 22, Paul Ross, Traffic Dynamics, pp 421-435, Copyright (1988), with permission from Elsevier

4.6 Ross Scenario 3

Ross scenario 3 is defined as follows [Ross 1988]:

a = 13 miles,

T = 0.2 hours = 12 minutes,

n = 1 lane for $1 \leq x \leq 13$,

$D_b = 950$ *vph,*

$S_b = 2000$ *vph,*

$k_{init} = 800/v_{free}$

Special Condition: The signal located at x = 5 miles is assumed to turn red at start of observation. It means that capacity is initially 0 at time *t=0.0 hr* and alternates between 0 and 2000 vph every 36 seconds.

Simulation Parameters: $\Delta x = 0.000625$ *miles,* $\Delta t = 6.25E\text{-}6$ *hrs*

The mathematical version of Scenario 3 is to use Godunov method, with $\Delta x=0.000625$ miles and $\Delta t=6.25E\text{-}6$ hours, to obtain a computational approximation to the (entropy) solution of the partial differential equation (12), for $0 \leq x \leq 13$ miles and $0 \leq \tau \leq 0.2$ hour, with flow function Q defined as in (26). Further, the initial condition is $k(x,0) = 800/63$ vehicle per mile, the entrant boundary condition at $x = 0$ is (21) with $D_b(\tau) \equiv 950$ vph for $0 \leq t \leq 0.2$ hour, the exit boundary condition at $x = a = 13$ *miles* is (22) with $S_b(\tau) \equiv 2000$ vph, and the point constriction condition of subsection (3.6) is to hold at $x = 5$ miles for $0 \leq t \leq 0.2$ hrs with the capacity of the link at $x = 5$ miles alternating between 0 and 2000 every 36 seconds starting with 0 at $\tau = 0$, with capacity as defined in section (1.1.2).

Ross described this scenario [Ross 1988] as "equilibrium speed-density formulations cannot model interrupted flow; such models essentially ignore the short section with capacity = 0 that is inherent in the signal scenario." This, if were to be true, would cast a very serious doubt over the utility of KWM models, because modeling signalized intersection is perhaps one of the basic problems that any valid traffic model should be able to simulate.

To discuss that solution analytically, we treat this problem as a point constriction (section 3.6) problem. At the beginning, the signal turns red and a point constriction of zero capacity is created at the five mile boundary. Let us assume that at this point of time, the vehicular density be $k_{sec} < 2000/63$ in the upstream link. Due to creation of point constriction, a shock wave is created connecting the upstream density of k_{sec} to the interface density $k_{red\text{-}\tau\text{-}}$. At point constriction, this density is equal to the jam density. Now, this wave travels upstream with a speed depending on the upstream density. On the downstream side, we have a shock wave traveling downstream with free flow speed connecting the upstream density, $k_{red\text{-}\tau+} = 0$ to the downstream density k_{down}. The shock waves are clearly seen in Figure 11. The upstream shock wave is the front left side of the plateau, facing the reader. The downstream shock is seen as the ripple generated at the signal boundary and propagating downstream.

When the light turns green, this situation can again be approximated with a point constriction having upstream density equal to k_{jam} and downstream density equal to zero. This creates a capacity controlled situation which is between the supply controlled scenario (section 3.5.2) and demand controlled scenario (section 3.5.3). This creates an acceleration wave which connects the upstream density of k_{jam} with $k_{green\text{-}\tau\text{-}} = 2000/63$. This acceleration wave travels faster than the preceding upstream shock wave because $k_{sec} < k_{max}$. The downstream shock connects upstream density of $k_{green\text{-}\tau+} = 2000/63$ *vpm* with the downstream density of *0 vpm*. The upstream acceleration wave can be clearly seen in Figure 11 as the back left side of the plateaus.

The state of traffic at the next red signal depends on the interaction between the shock wave generated during preceding red signal and the acceleration wave generated during the green phase. When the acceleration wave overtakes the shock wave, as it is moving faster than it, the only remaining wave is a wave connecting the upstream density of k_{sec} to the downstream density of k_{max}. As k_{max} is greater than k_{sec}, this leads to generation of a shock wave traveling downstream with free flow speed. If the shock wave is able to reach the signal, it would lead to a demand starved (section 3.5.3) scenario else it would a capacity limited situation. The results can easily be matched with the Ross results depicted in Figure 13.

A better appreciation of the results can be done by looking at the cell just upstream of the signal. This is shown in Figure 12. We see that the density rapidly increases to the jam density as soon as the signal turns red. The density remains equal to jam density until the signal turns green. As soon as the signal turns green, we see the drop in density to $k_{max} = 2000/\ 63 \approx 31.7\ vpm$. What is interesting to note is the further drop in the observed density. This can be attributed to the phenomenon discussed in the preceding paragraph. In this case, the final shock wave, caused due to merger of the red signal generated shock wave and green signal generated acceleration wave, arrives at the signal. Due to this the density decreases to ≈ 15 vpm. Another observation is that the troughs are initially wider and slowly get narrower. This can be the effect of the increased demand from 800 vph to 950 vph finally reaching the signal.

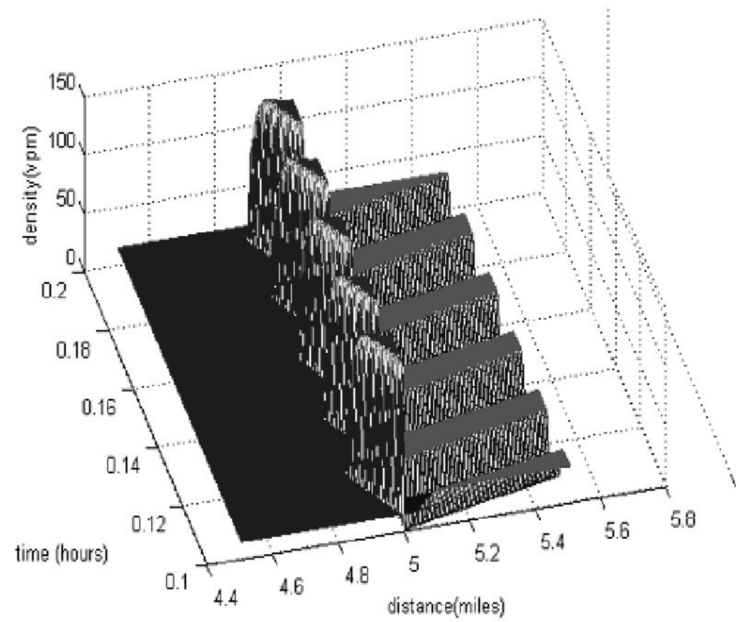


Figure 11. Simulation result for Ross scenario 3 (signalized intersection)

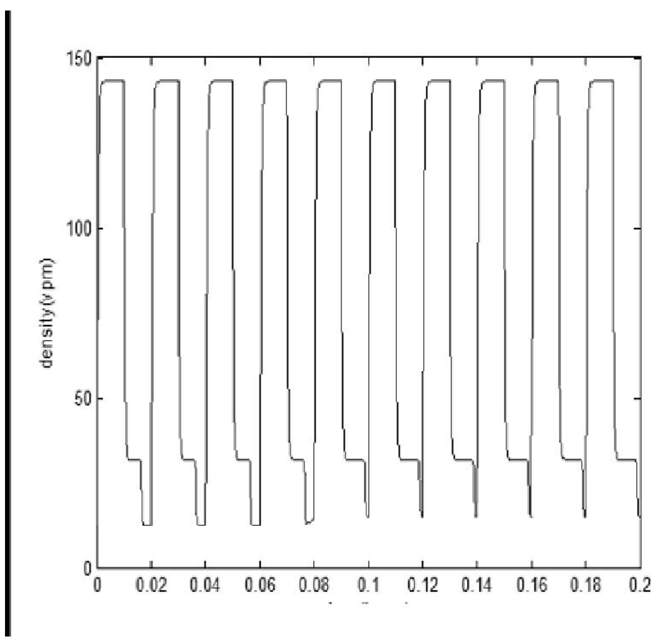


Figure 12. Density vs time at link just upstream of the signal

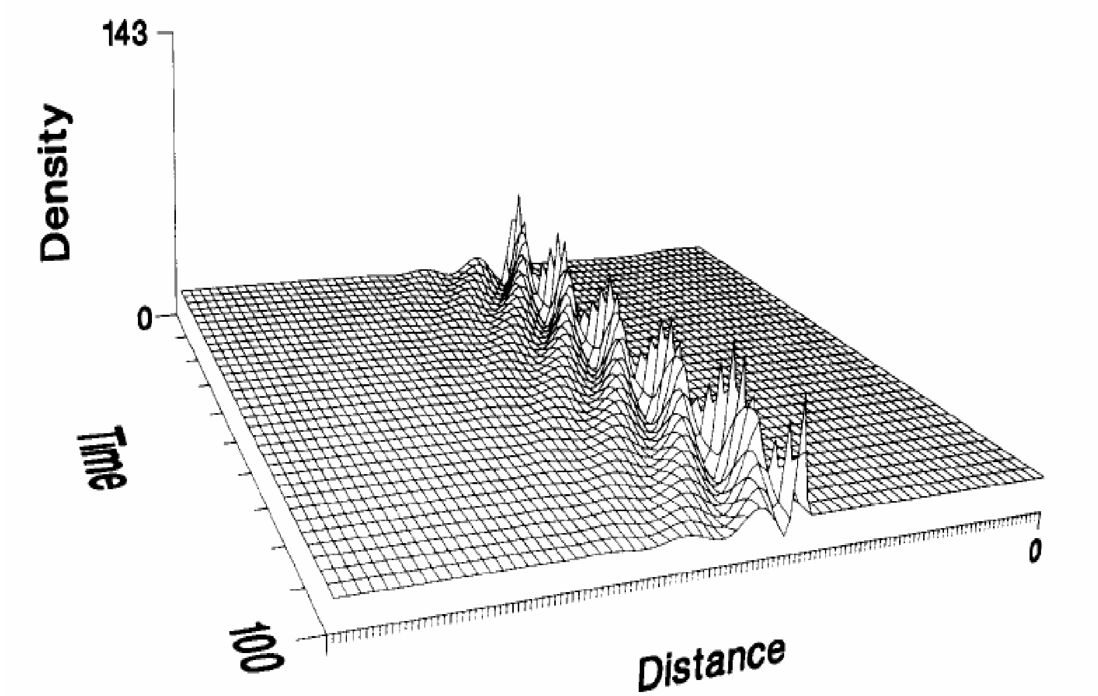


Figure 13. Result reported by Ross for scenario 3. Reprinted from Transportation Research B, Vol 22, Paul Ross, Traffic Dynamics, pp 421-435, Copyright (1988), with permission from Elsevier

The simulation results and their comparison with the original Ross results, given in Figure 13, confirm that combination of Godunov method, triangular TSM and correct boundary, interface and point constriction analysis comes very close to reproducing the Ross scenarios. This is an affirmation of the validity of the KWM model. In the next chapter we will look at the actual simulation details and the possible extensions to that.

CHAPTER V

LWR SIMULATION

In Chapter IV, we compared the results obtained for Ross scenarios using the simulation against the results reported by Ross and analytical results. These particular scenarios were chosen for two main reasons: The first reason is that these scenarios provide diverse scenarios which for which intuitive traffic behavior can be obtained. The other reason was because Ross, in the paper [Ross 1988] had cast serious doubts regarding the capability of LWR model to accurately predict the scenario results. This paper asserted that LWR was not capable of simulating even the most basic of traffic scenarios. This software simulates these basic scenarios as suggested by Ross. This chapter looks in detail at the various components of this simulation software.

5.1 Design Goals

The initial goal of this simulation was to be able to simulate the Ross scenarios. Apart from being able to simulate these specific scenarios, one other important goal was to create a modularized version of software so that it can be easily extended to accommodate more generic cases in the future. The software is finally intended to be used by two separate sets of users. The first are the *casual users* who are not very familiar with the technical details and just want to change a few parameters and see the changes in result. The other set of users, *technical users*, would be those who understand the software and would make significant changes to it. The changes could be as small as changing few parameters or could be as widespread as adding scenarios to the simulation. Hence, one of the biggest design challenges was to balance the needs of these diverse groups.

5.2 Important Modules

The simulation is done in MATLAB® 6.5. There are two distinct tiers in this software. The first is presentation tier or the user interface tier while the second tier is the actual implementation logic tier. The presentation tier is created to make the simulation easy to use with minimum effort required from the end user. This is the tier that contains the GUI screens and the property files. Although, strictly speaking, property files should be a part of implementation layer, they are considered a part of presentation layer as they are exposed to the end user. The implementation logic tier is responsible for actually implementing the different scenarios. This tier is not exposed to the end users. This is done so that the actual implementation logic can be changed without affecting the end user.

All the presentation tier screens are designed such that all the future extensions can be done with minimal effort. The main screens for the presentation tier are as follows:

5.2.1 Welcome Screen

This is the first screen shown to the user when the simulation is launched (see APPENDIX A). Besides being the welcome screen, it helps the user to choose the scenario to be simulated at a very high level. At present the simulation is specific to Ross scenarios, so there are only two choices provided. The first is "*Ross scenarios*" while all the other choices are grouped under "*Others*". As we add more and more scenarios, we can make other groups. The selections are provided in form of radio buttons, so the user has the option of selecting from a group of scenarios. The help buttons on the screen provide context sensitive information, which is then shown in a different page. There are two other buttons on the screen. The "*Start*" button shows the details of the selected scenario group, while the "*Exit*" button is used to close the simulation. The welcome screen is shown in Figure 14.

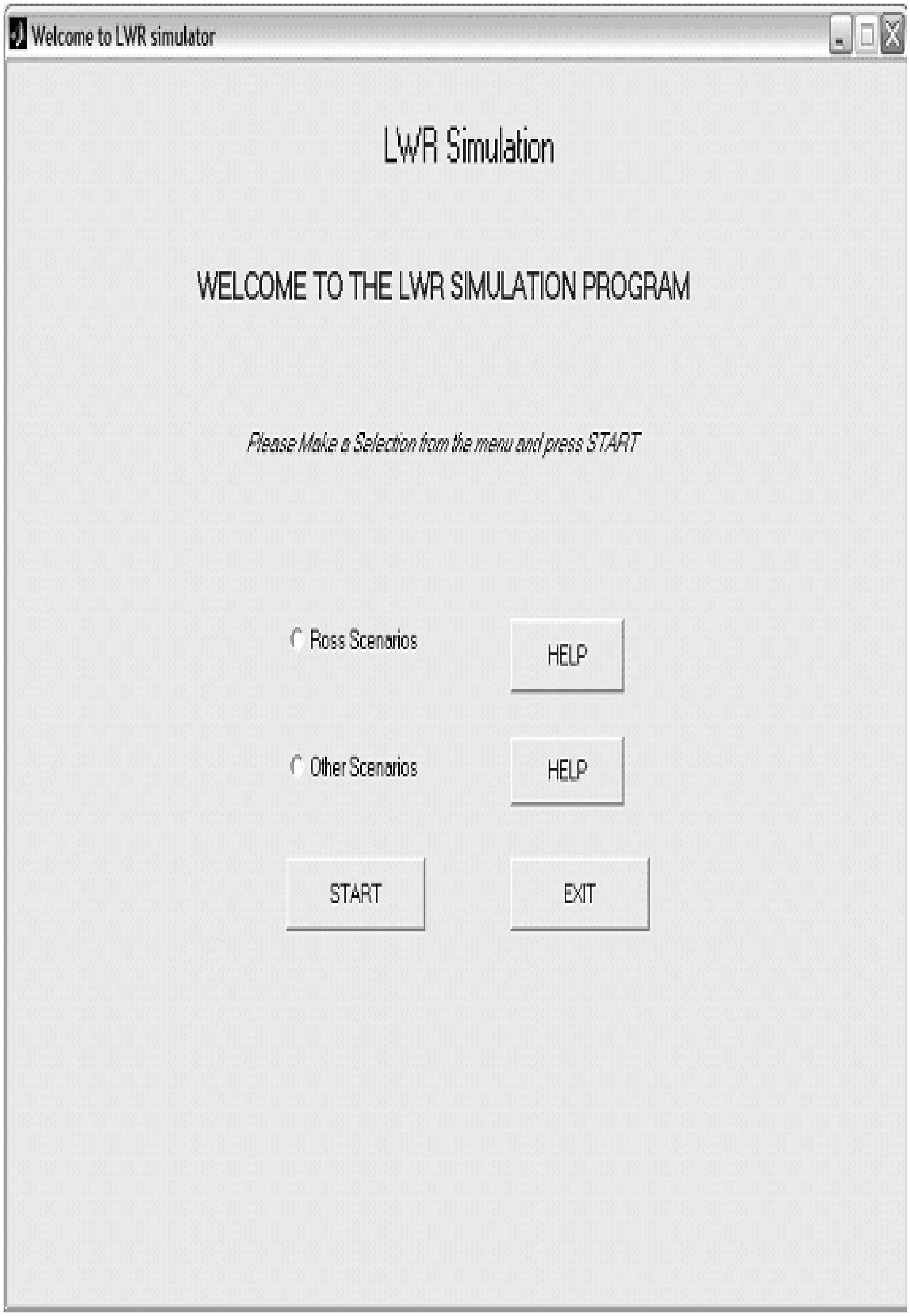


Figure 14. The welcome screen

5.2.2 Detail Screen

This screen is displayed when the user selects a specific category of simulation. In the present scenario, these categories are *Ross Scenarios* and *Others*. The detail screen allows the user to select a specific scenario from the general set of scenarios and also provide various parameters associated with the chosen scenario. In the present Ross scenarios screen (Figure 15), the user can choose from lane drop, modified lane drop, signalized intersection and accident. The user can then specify the important parameters and press the "*Simulate*" button to run the simulation. The software then takes all the initially defined default values and generates a 3-D figure showing vehicular density as a function of time and space (Figure 16). An important property of this screen is the error checking done on for the user provided parameters. This helps the user to detect common errors before they are passed to the backend to be computed. Users are shown error message in pop up boxes (Figure 17) which pass the information to the user who can then correct the error. Another important consideration was to make the software very easy to understand. This is achieved by providing context sensitive help. The user can invoke the help screens at any point by clicking the help button located next to the parameters (Figure 18).

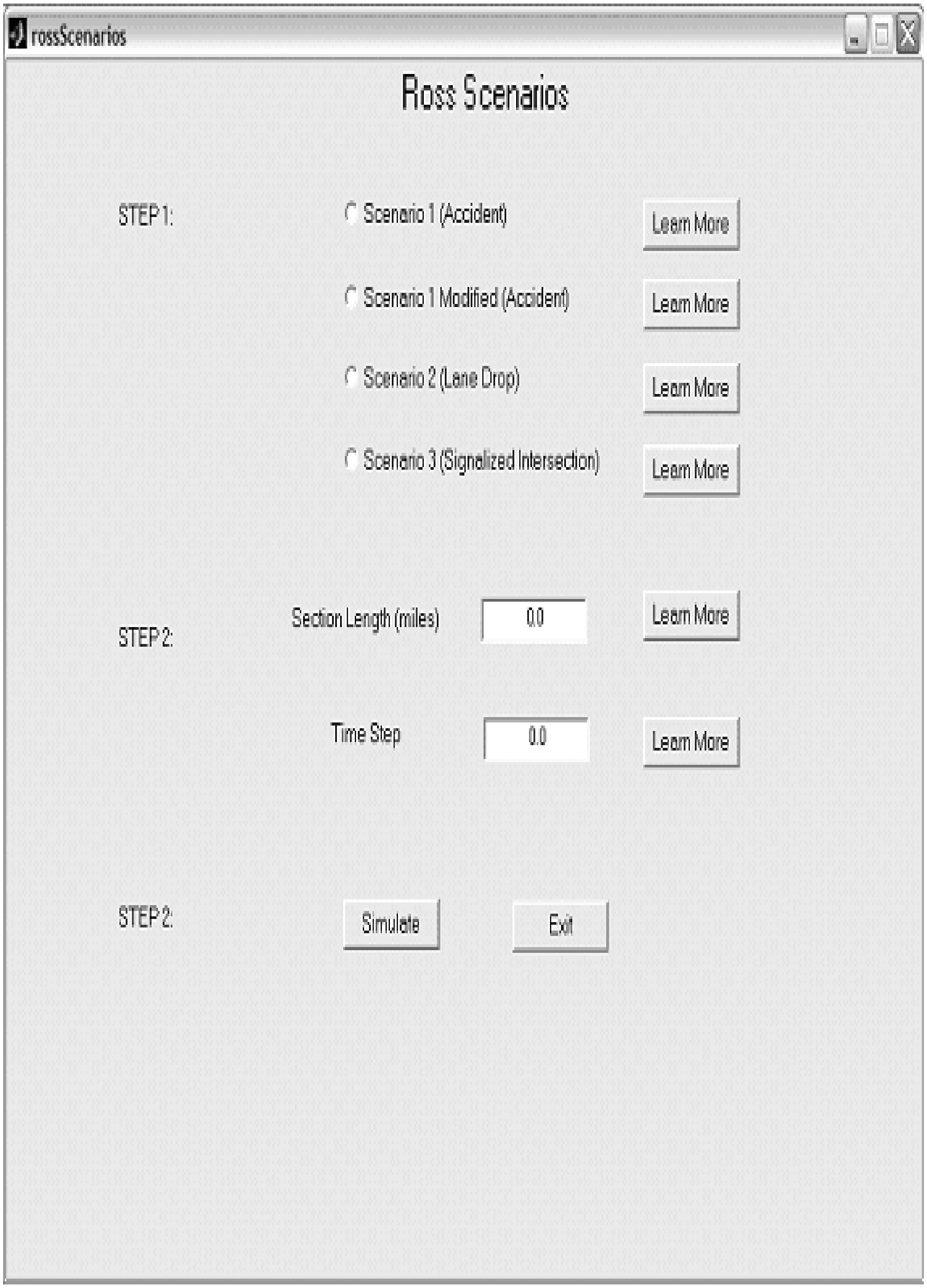


Figure 15. Ross scenario detail screen

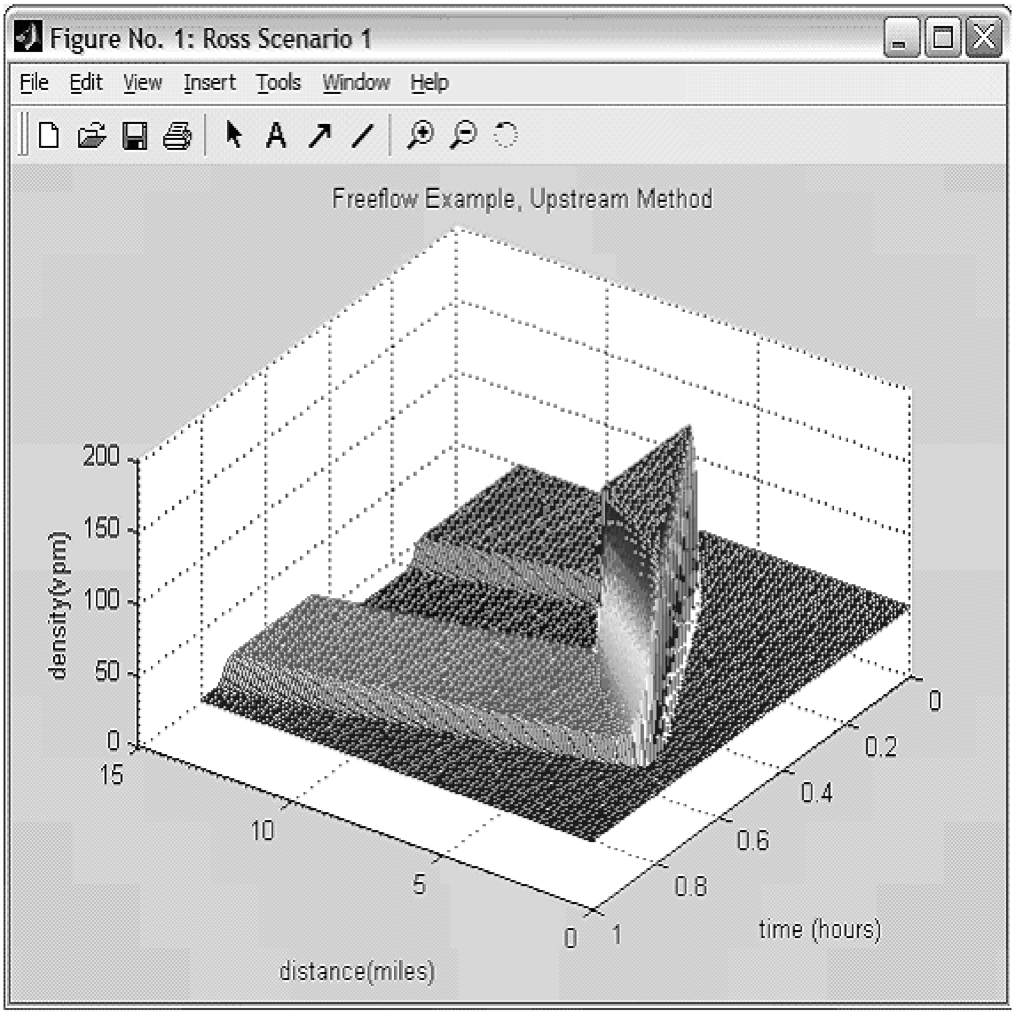


Figure 16. Sample simulation output

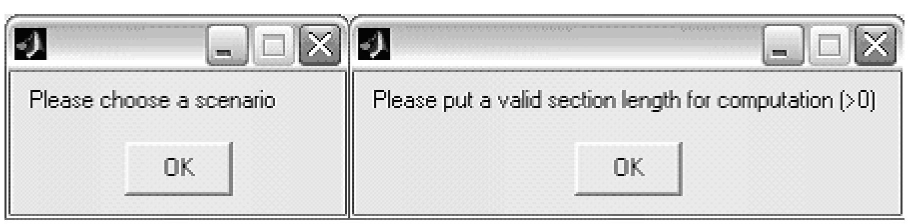


Figure 17. Sample error messages

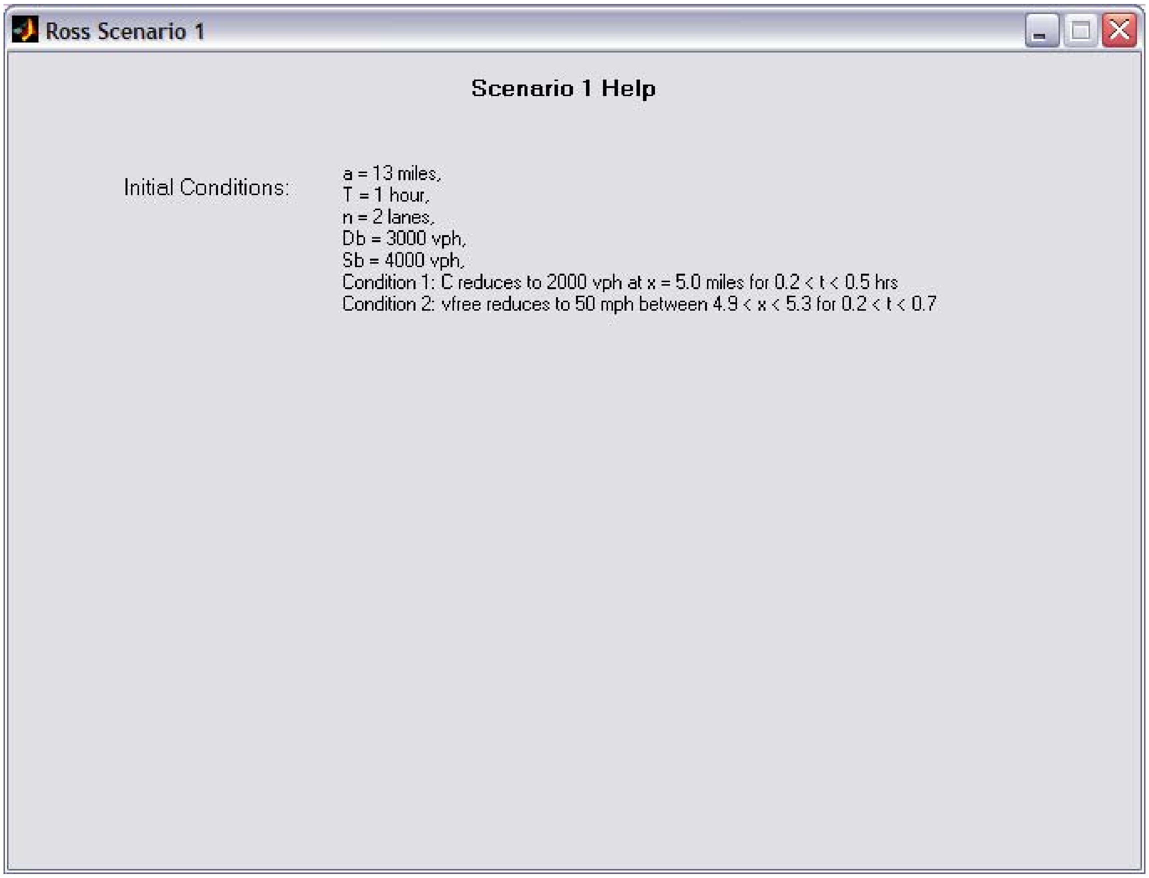


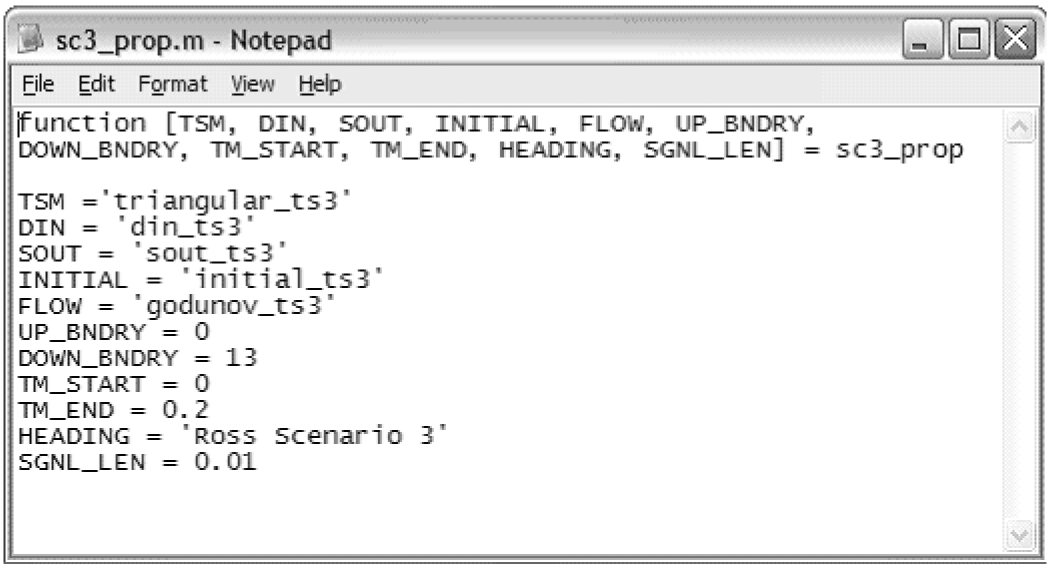
Figure 18. Sample help screen

5.2.3 Property File[1]

This file is the link between the presentation layer and the implementation layer. Although the *casual* user would not normally modify this file in normal running, a researcher might like to frequently change this file, as it contains all the default parameters. This structure was chosen because it gives flexibility both in terms of end user and expert user. An end user wants to change minimum number of parameters and also wants to keep the GUI as simple as possible letting the software decide the default values. On the other hand, an expert user would like to change lots of parameters but is not very interested in GUI. Property file provides this advantage. Most of the default

[1] The term 'Property File' is used in generic sense as a file containing list of parameters in the *name=value* format.

values are contained in the property file. Another flexibility that property files provide is the ability to change parameters without restarting the program. Hence, the expert user can just change the parameters and press the "*Simulate*" button once again. Sample property file for Ross scenario 3 is shown in Figure 19.


sc3_prop.m - Notepad

File Edit Format View Help

```
function [TSM, DIN, SOUT, INITIAL, FLOW, UP_BNDRY,
DOWN_BNDRY, TM_START, TM_END, HEADING, SGNL_LEN] = sc3_prop

TSM ='triangular_ts3'
DIN = 'din_ts3'
SOUT = 'sout_ts3'
INITIAL = 'initial_ts3'
FLOW = 'godunov_ts3'
UP_BNDRY = 0
DOWN_BNDRY = 13
TM_START = 0
TM_END = 0.2
HEADING = 'Ross Scenario 3'
SGNL_LEN = 0.01
```

Figure 19. Sample property file

5.2.4 Implementation Logic Files

In order to keep the program as modular as possible, the implementation logic is split into many small files. This makes the program easy to maintain but more importantly it breaks the program into smaller chunks so that modifications can be made locally without having to change many files. For the Ross scenarios, the major files are the following (Appendix B):

lwr.m: This is the main driver file. It reads the property file and sets up all the default functions and values. This also calls the various other functions to compute the vehicular density as function of space and time. Finally it is responsible for generating the 3-D figure depicting the result.

triangular.m: This contains the implementation of the traffic stream model employed to compute the flow from a link. If at a later stage the TSM needs to be changed, the only change needed is specifying the new TSM in the property file because the definition of TSM to be used is present only in property file. Hence, this makes changes very easy to make.

godunov.m: This contains the implementation for the Godunov method. This function is also responsible for calling the appropriate functions for computing the demand and supply in a link and then returning the resultant flow from the link.

demand.m: This contains the implementation for evaluating the upstream demand at any link, given upstream density, distance from the origin and the time. The exception is the starting link, whose demand is determined through the function *din*.

supply.m: This contains the implementation for evaluating the downstream supply at any link, given downstream density, distance from the origin and the time. The exception is the last link, whose supply is determined through the function *sout*.

initial.m: This function is responsible to set up the initial conditions along the roadway. More specifically, it sets the initial vehicular density along the roadway.

CHAPTER VI

CONCLUSION

The main objective of this thesis was to simulate the various scenarios presented by Ross [1988], with a view toward verifying the ability of the KWM to reproduce expected traffic behavior for those scenarios when boundary conditions, interface conditions and point constrictions are treated as described in Chapter III. These “Ross scenarios” include a lane drop, an accident and behavior of a signalized intersection.

The simulation was created in MATLAB® 6.5, using a triangular traffic stream model. The details of the results thus obtained are explained in detail in Chapter IV. These generated results are qualitatively and quantitatively very close to those expected analytically which in turn reproduces much of how one would intuitively expect traffic to behave under the respective scenarios. This verifies the correctness of the proposed (Chapter III) numerical treatment of boundary and interface conditions and point constrictions and provides a modest degree of validation of the KWM.. Although a lot of work is still to be done to make it a viable tool to analyze and simulate complicated traffic scenarios, this simulation serves as a proof of concept.

6.1 Future Work

The simulation software created is presently tailored to simulate the Ross scenarios. The future work could include providing extensions to the software which would make it generic enough to encompass more complicated traffic scenarios. Although the present software does not cater to this, the design of the GUI has been done keeping such extensions in mind. The proposed future work can be divided into the following subsections.

6.1.1 Functionality Related Enhancement

At present the simulation is concerned only with validating Ross scenarios. This, although important, is still very restrictive. The future version of simulation can provide more cases for user to choose from. An example of such extension is provided in the "Other Scenario" section of the welcome screen of the simulation GUI. The user can also be given the choice of choosing from a number of TSM, thus providing the ability to compare and evaluate them.

At present the results are provided in form of a 3-D figure. Although this provides an intuitive method of analyzing the results, this can be further improved by providing the user with an option of generating a result in a known format for e.g. HCM 2000 format or in form of a movie.

6.1.2 UI Related Enhancement

At present most of the user input is given through properties file. Although this has the advantage of being flexible and easily modifiable solution, this still requires the user to manually modify the property file which in turn also requires the user to be aware of the property file. This is perhaps acceptable for users who understand the software well, but can be a hindrance for end users who would like to work within the software screens. For this, in future versions, the default property file can be shown to the user in the GUI itself and any modifications made can then be updated in the file automatically.

REFERENCES

CASSIDY, M. J. 1998. Bivariate relations in nearly stationary highway traffic. *Transportation Research B. 32*, *1*, 49-59.

COHEN, S.L. 2002. Application of Car-Following Systems in Microscopic Time-Scan Simulation Models. *Transportation Research Record 1802.* paper No. 02-2759. pp. 205-213.

DAGANZO, C. F. 1995. The cell transmission model, part II: network traffic. *Transportation Research B. 29*, 79-93.

DAGANZO, C. F., MUÑOZ, J. C. 2002. The bottleneck mechanism of a freeway diverge. *Transportation Research Part A. 36*, 483-505.

DRAKE, J. S., SCHOFER, J. L., AND MAY, A. D. 1967. A statistical analysis of speed density hypothesis, *Highway Research Record 154*. Washington, D.C. pp. 23-87.

GREENSHIELDS. B. N. 1934. A study of traffic capacity. In *Proceedings of the Annual Meeting –Highway Research Board*. pp. 448-474.

KOSHI, M., IWASAKI, M., AND OHKURA, I. 1983. Some findings and an overview on vehicular flow characteristics. In *Proceedings on the Eighth International Symposium on Transportation and Traffic Theory.* V. Hurdle, E. Hauer, G. Stuart Eds. Elsevier. Amsterdam. pp. 403-426.

KUHNE, R., AND MICHALOPOULOS, P. 2002. Continuum flow models, *Revised Monograph on Traffic Flow Theory.* URL: http://www.tfhrc.gov/its/tft/tft.htm. Last Modified: 05/07/2002.

LEBACQUE, J. P. 2003. Intersection modeling, application to macroscopic network traffic flow models and traffic management. *Unpublished publication*.

LEBACQUE, J.P. 1996. The Godunov Scheme and what it means for first order traffic flow models. In *Transportation and Traffic Theory: Proceedings of the 13th ISTTT.* J.B. Lesort, ED. Pergamon, Oxford. pp. 647-677.

LIGHTHILL, M. J., AND WHITHAM, G.B. 1955. On kinematic waves II - A theory of traffic flow on long crowded roads. *Proc. Royal Society, London. A229*, 317-345.

MICHALOPOULOS, P. G., BESKOS, D.E., AND LIN, J.K. 1984. Analysis of interrupted traffic flow by finite difference methods. *Transportation Research Board. 18*, 409-421.

NELSON, P., KUMAR, N.2004. Point Constriction, Interface and Boundary Conditions for kinematic-wave model. *Transportation Research Board.* URL: http://www.trb.org/news/blurb_detail.asp?id=1121. Last Modified: 01/12/2004

NEWELL, G. F. 1989. Comments on Traffic Dynamics. *Transportation Research B. 23B*, 386-389.

NEWELL, G. F., A Simplified theory of kinematic waves in highway traffic, Part II: Queueing at freeway bottlenecks, *Transportation Research B*, *29*, *1995*, 261-276.

PAPAGEORGIOU, M. 1998. Some remarks on macroscopic traffic flow modeling. *Transportation Research A. 32*, *5*, 323-329.

PAYNE, H.J. 1971. Models of freeway traffic and control. In *Simulation Council Proceedings. 1* , *1*, 51-61.

RICHARDS, P. I. 1956. Shockwaves on the highway. *Operations Research*, *4*, 42-51.

ROSS, P. 1988. Traffic dynamics. *Transportation Research B. 22*, 421-435.

WILLIAM, J.C. 2002. Macroscopic flow models. *Revised Monograph on Traffic Flow Theory*. . URL: http://www.tfhrc.gov/its/tft/tft.htm. Last Modified: 05/07/2002

APPENDIX A

This Appendix is concerned with the important components of the program as seen from the perspective of *casual* users. This includes the GUI screens. These screens are illustrated by running through Ross scenario 1, highlighting the important parts.

Welcome Screen

In this screen the *casual* user selects the group of simulation that he is interested. In this case he chooses Ross scenarios. This is done by selecting the appropriate radio button (Figure 20). There are three other actions that the user can perform on this screen. The first is getting help on the groups, second is moving on to the next screen by clicking on the *Start* button and the third is exiting the application by clicking the *End* button.

If the user is wants to know more about the group, he can click on any of the *Help* buttons. A help screen is shown with more information about the corresponding group. For e.g. clicking on the *Help* for the *Ross Scenario* group shows Figure 21.

If the user wants to exit the application at this point, he can click on the *End* button. This closes the welcome screen.

If the user is sure about the group, he can select the group and click on the *Start* button. This launches another screen (Figure 15) with details about all the scenarios in the group and the related parameters. In case the user has not selected any group and clicks the *Start* button, an error message is displayed (Figure 22).

In our case, the user selects the *Ross Scenario* group and clicks *Start*.

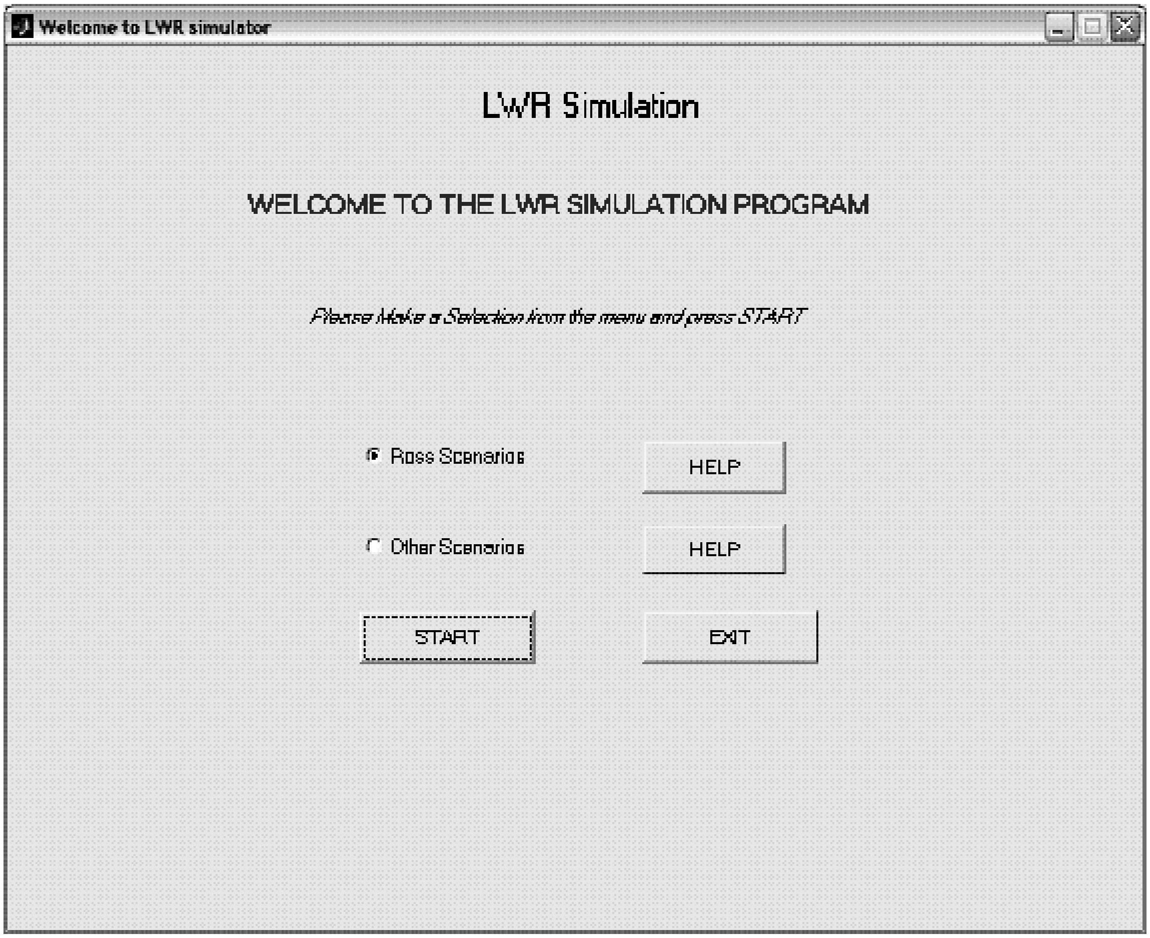


Figure 20. Welcome screen

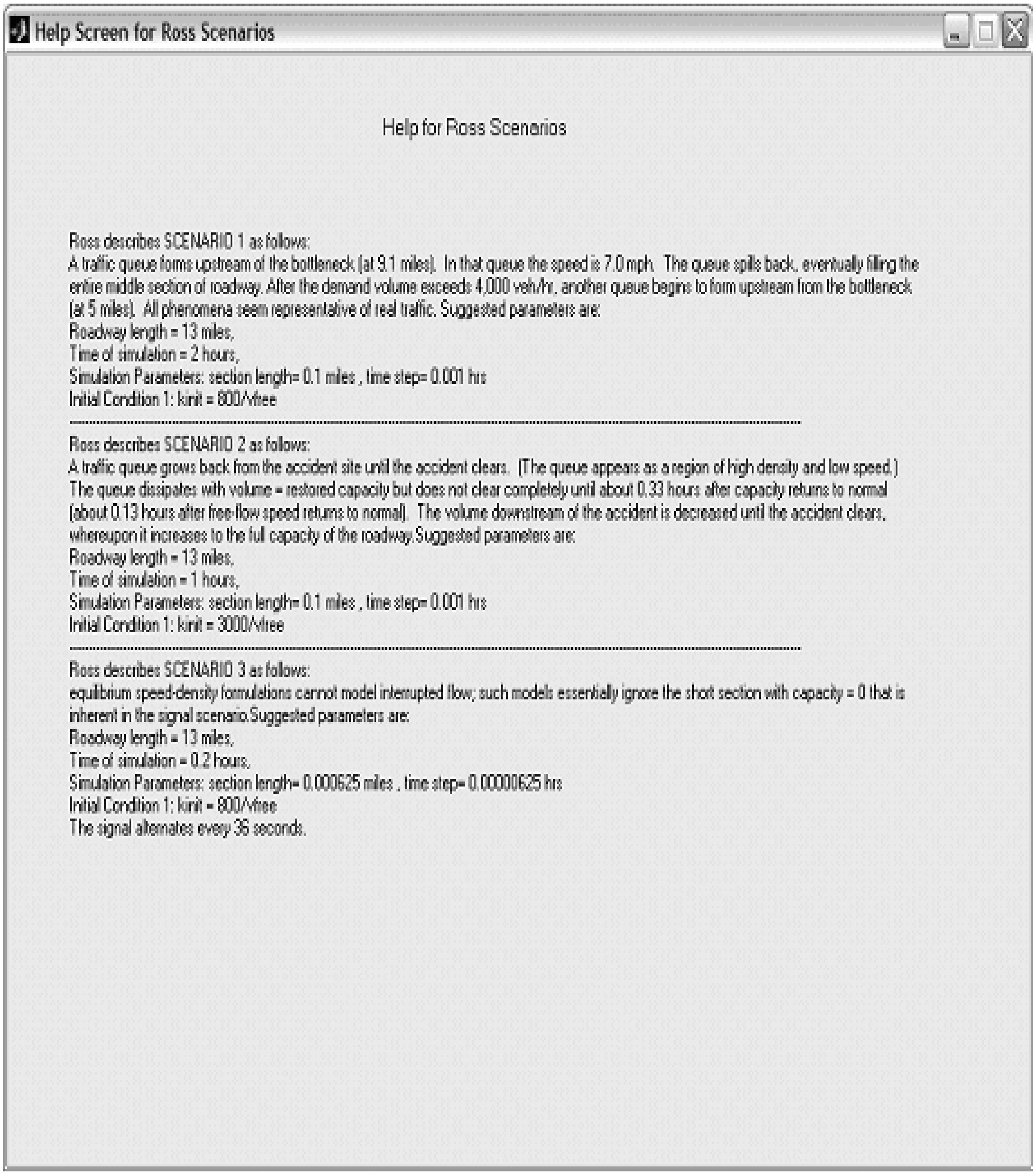


Figure 21. Sample help screen (simulation groups)

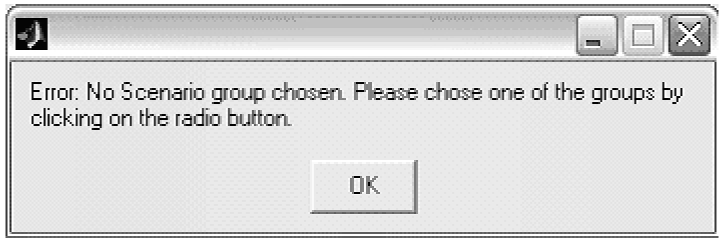


Figure 22. Error message

Detail Screen

This screen (Figure 15) is shown to the user once he has selected *Ross Scenario* group. In this screen, we presently have three Ross Scenarios and modified version of Ross scenario 1. The screen is divided into three parts: In the first part, user selects the specific scenario, in the second part, the user enters the parameters specific to the scenario and in the third part user actually starts the simulation.

In the first step the user select a specific scenario to simulate. In the present case, the selection can be done from the three Ross scenarios and one modified scenarios. If the user is confused about any scenario or wants to know more about a specific scenario, he can click on any of the *Learn More* buttons. A help screen is shown with more information about the corresponding scenario. For e.g. clicking on the *Learn More* for the *Ross Scenario 1* displays Figure 18.

In the second step, the user provides parameters required to be set for the simulation. In the present case, the link length and the time step duration is specified (Figure 23).

In the third step, user can either start the simulation by clicking the *Simulate* button or can exit this screen by clicking on the *Exit* button.

Another important feature of the screen is the error checks. This is important for the *casual* users so that they do not have to worry about passing incorrect or insufficient information to program to process. For e.g. if the user has entered a negative value for time step or link length or has forgotten to select a scenario, he is prompted with an error message to correct it. The error message is self explanatory in nature so that user knows exactly where the error has occurred and the possible solution for it. One sample error message is given in Figure 24.

Once the user clicks the *Simulate* button, the program computes the vehicular density as a function of distance and time. The result is then displayed in a 3-D figure. In our case, Figure 5 gives the program output for *Ross Scenario 1* with the *Link Length = 0.1 miles* and *Time Step = 0.001 hours*. The user can then change the parameters and generate another figure and compare them to analyze the effect of changing one or more parameters.

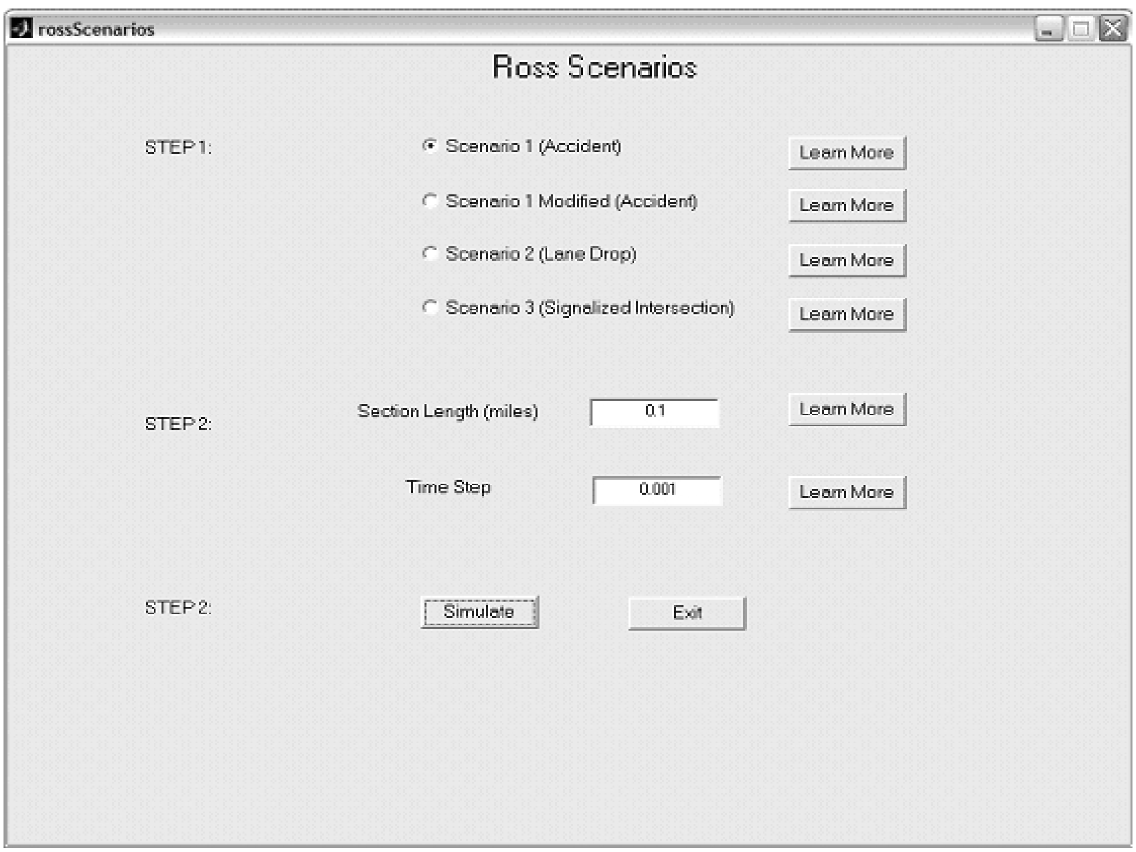


Figure 23. Detail screen with parameters filled in for scenario 1

Figure 24. Sample error message

APPENDIX B

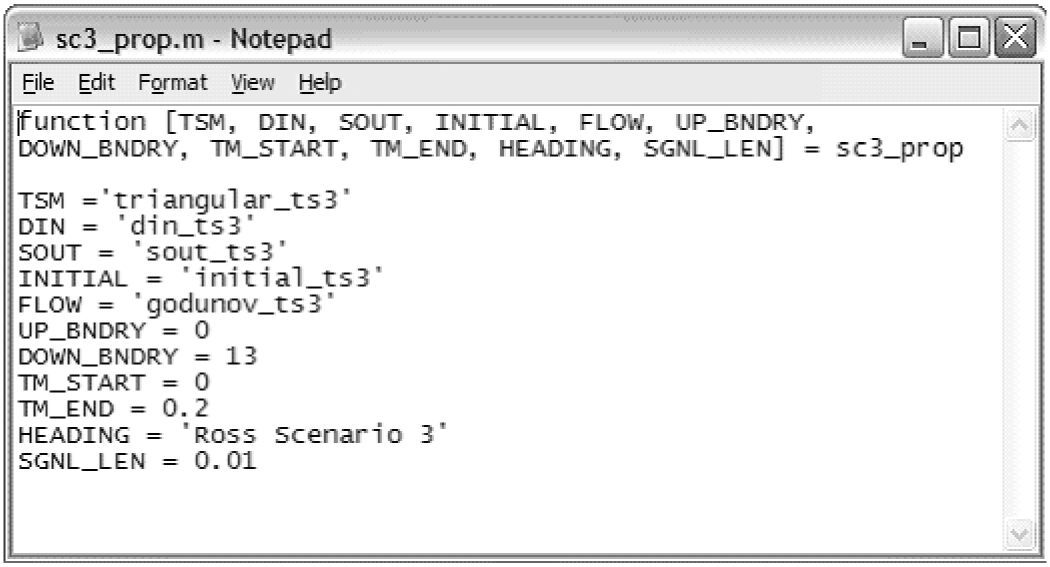
This Appendix is concerned with the important components of the program as seen from the perspective of *technical* users. This includes the GUI screens, property file and results. These components are illustrated by running through same Ross scenario as *casual* users, highlighting the important differences.

GUI Components

The GUI components used by the technical users are same as those used by *casual* users (refer to Appendix A for more details).

Property File

This file (Figure 25) is perhaps the most important file for the *technical* user. This also serves as the link between the presentation layer and the implementation layer. This file contains list of parameters for the simulation being run. This property file can be used by the user to change parameters like time of simulation, length of roadway, heading etc., which are used in the simulation. The file can also be used to bring about massive changes like replacing the TSM or changing the numerical scheme itself. Hence, this file can be used very effectively by a *technical* user.




```
function [TSM, DIN, SOUT, INITIAL, FLOW, UP_BNDRY,
DOWN_BNDRY, TM_START, TM_END, HEADING, SGNL_LEN] = sc3_prop

TSM ='triangular_ts3'
DIN = 'din_ts3'
SOUT = 'sout_ts3'
INITIAL = 'initial_ts3'
FLOW = 'godunov_ts3'
UP_BNDRY = 0
DOWN_BNDRY = 13
TM_START = 0
TM_END = 0.2
HEADING = 'Ross Scenario 3'
SGNL_LEN = 0.01
```

Figure 25. Property file for scenario 2

APPENDIX C

This Appendix provides the code for simulation of Ross Scenario 2.

appStart.m

```
function varargout = appStart(varargin)
% APPSTART M-file for appStart.fig
%      To launch this figure, type APPSTART at MATLAB prompt
% This is the welcome screen which is shown to the user when the
% application is launched. Besides being the welcome screen, this screen
% also provides user with option of choosing the group of scenarios to be
% simulated
%
% Version History:
% Nishant Kumar     27-May-2004     Added comments

% Following is the internal GUIDE code which sets up the environment
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @appStart_OpeningFcn, ...
                   'gui_OutputFcn',  @appStart_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin & isstr(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% This function executes just before the form is made visible to the user.
% This is the place where all the default values are set.
% This function has no output args, see OutputFcn.
% Input Params: hObject    handle to figure
%               eventdata  reserved - to be defined in a future version of MATLAB
%               handles    structure with handles and user data (see GUIDATA)
```

```
%              varargin   command line arguments to appStart (see VARARGIN)
% Version History:
% Nishant Kumar     27-May-2004     Added code for initializing default
%                            group
function appStart_OpeningFcn(hObject, eventdata, handles, varargin)

% Choose default command line output for appStart
handles.output = hObject;

% This section sets all the default values
handles.group_selected = 0;  % this is the default selection. Initially set to 0

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
% Output params:    varargout  cell array for returning output args (see
VARARGOUT);
% Input params:     hObject    handle to figure
%                eventdata  reserved - to be defined in a future version of MATLAB
%                handles    structure with handles and user data (see GUIDATA)
function varargout = appStart_OutputFcn(hObject, eventdata, handles)

% Get default command line output from handles structure
varargout{1} = handles.output;

% This function is called when the user selects the Ross Scenario group.
% This function then toggles the radio button if there was any earlier
% selection and sets the group selected to "Ross Scenario".
function rossScenarios_Callback(hObject, eventdata, handles)
   % set the other radio buttons to off
   hdlList = [handles.otherScenarios];
   toggleRadioBtns(hdlList);
   handles.group_selected = 1;
   guidata(hObject, handles);


% This function is called when the user selects the Others group.
% This function then toggles the radio button if there was any earlier
% selection and sets the group selected to "Others".
function otherScenarios_Callback(hObject, eventdata, handles)
   % set the other radio buttons to off
   hdlList = [handles.rossScenarios];
   toggleRadioBtns(hdlList);
```

```
    handles.group_selected = 2;
    guidata(hObject, handles);


% This function is called when the user presses the "Start" button after
% selecting the group. It then selects and opens the appropriate detail
% screen.
function btnStart_Callback(hObject, eventdata, handles)
    % This link gets the chosen group from the global parameters. If
    % there was no group chosen, then an error message box is displayed.
    chosenGroup = handles.group_selected;
    if (chosenGroup == 0)
        msgbox('Error: No Scenario group chosen. Please chose one of the groups
by clicking on the radio button.');
        return;
    end
    switch chosenGroup
        case 1
            openfig(rossScenarios,'reuse');
        case 2
            openfig(otherScenarios,'reuse');
    end


% This function closes the present form.
function btnExit_Callback(hObject, eventdata, handles)
    delete(appStart);

% This function sets the value attribute of the array of objects being
% passed in to "off". This function would be used here to ensure that
% mutual exclusion is obtained in the radio buttons.
% input params:
% btnList      Handles of radio buttons.
function toggleRadioBtns(btnList)
    set(btnList, 'Value', 0);


% This function launches the help function for other.
function bHelpOther_Callback(hObject, eventdata, handles)
    openfig(help_otherScenario,'reuse');

% This function launches the help function for Ross scenarios.
function bHelpRoss_Callback(hObject, eventdata, handles)
    openfig(help_rossScenario,'reuse');
```

rossScenarios..m

```
function varargout = rossScenarios(varargin)
% ROSSSCENARIOS M-file for rossScenarios.fig
% To launch this figure type ROSSSCENARIOS
%
% Version History:
% Nishant Kumar     27-May-2004     Added comments

% Following is the internal GUIDE code which sets up the environment
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @rossScenarios_OpeningFcn, ...
                   'gui_OutputFcn',  @rossScenarios_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin & isstr(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% This function executes just before the form is made visible to the user.
% This is the place where all the default values are set.
% This function has no output args, see OutputFcn.
% Input Params: hObject    handle to figure
%               eventdata  reserved - to be defined in a future version of MATLAB
%               handles    structure with handles and user data (see GUIDATA)
%               varargin   command line arguments to appStart (see VARARGIN)
% Version History:
% Nishant Kumar     27-May-2004     Added code for initializing default
%                                   scenario selection
function rossScenarios_OpeningFcn(hObject, eventdata, handles, varargin)
```

```
% Choose default command line output for rossScenarios
handles.output = hObject;

% here the variables are being initialized
handles.scenario_selected = 0;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
% Output params:    varargout  cell array for returning output args (see
VARARGOUT);
% Input params:     hObject    handle to figure
%                   eventdata  reserved - to be defined in a future version of MATLAB
%                   handles    structure with handles and user data (see GUIDATA)
function varargout = rossScenarios_OutputFcn(hObject, eventdata, handles)
   varargout{1} = handles.output;

% This function is called when the user selects the Ross Scenario 1.
% This function then toggles the radio button if there was any earlier
% selection and sets the group selected to "Ross Scenario 1" in the global list.
function rbScenario1_Callback(hObject, eventdata, handles)
   % set the other radio buttons to off
   hdlList = [handles.rbScenario2, handles.rbScenario3,
handles.rbScenario1Mod];
   toggleRadioBtns(hdlList);
   handles.scenario_selected = 1;
   guidata(hObject, handles);

% This function is called when the user selects the Ross Scenario 2.
% This function then toggles the radio button if there was any earlier
% selection and sets the group selected to "Ross Scenario 2" in the global list.
function rbScenario2_Callback(hObject, eventdata, handles)
   % set the other radio buttons to off
   hdlList = [handles.rbScenario1, handles.rbScenario3,
handles.rbScenario1Mod];
   toggleRadioBtns(hdlList);
   handles.scenario_selected = 2;
   guidata(hObject, handles);

% This function is called when the user selects the Ross Scenario 3.
% This function then toggles the radio button if there was any earlier
% selection and sets the group selected to "Ross Scenario 3" in the global list.
function rbScenario3_Callback(hObject, eventdata, handles)
```

```
    % set the other radio buttons to off
    hdlList = [handles.rbScenario2, handles.rbScenario1,
handles.rbScenario1Mod];
    toggleRadioBtns(hdlList);
    handles.scenario_selected = 3;
    guidata(hObject, handles);

% This function is called when the user selects the modified Ross Scenario 1.
% This function then toggles the radio button if there was any earlier
% selection and sets the group selected to "Modified Ross Scenario 1" in the
% global list.
function rbScenario1Mod_Callback(hObject, eventdata, handles)
    % set the other radio buttons to off
    hdlList = [handles.rbScenario2, handles.rbScenario1, handles.rbScenario3];
    toggleRadioBtns(hdlList);
    handles.scenario_selected = 4;
    guidata(hObject, handles);


% This function sets the value attribute of the array of objects being
% passed in to "off". This function would be used here to ensure that
% mutual exclusion is obtained in the radio buttons.
% btnList      Handles of radio buttons.
function toggleRadioBtns(btnList)
    set(btnList, 'Value', 0);

% This function opens up the help screen for scenario 1.
function bHelpSc1_Callback(hObject, eventdata, handles)
    openfig(help_scenario1, 'reuse');

% This function opens up the help screen for scenario 2.
function bHelpSc2_Callback(hObject, eventdata, handles)


% This function opens up the help screen for scenario 3.
function bHelpSc3_Callback(hObject, eventdata, handles)


% This function opens up the help screen for link length.
function bHelpSecLen_Callback(hObject, eventdata, handles)

% This function opens up the help screen for time step.
function bHelpTimeStep_Callback(hObject, eventdata, handles)
```

```
% This function closes the present form.
function bExit_Callback(hObject, eventdata, handles)
   delete(handles.rossScenarios);

% This function executes on button press in bSimulation. This function is
% the main function which is responsible for setting the appropriate parameters
% and then calling the function which evaluates all the parameters.
function bSimulation_Callback(hObject, eventdata, handles)
   % The first thing is to collect all the parameters selected by the
   % user. The first thing to get is the actual scenario selected. The
   % next thing is to get the link length and the time step

   chosenScenario = handles.scenario_selected;
   secLength = str2double(get(handles.tbSecLen, 'string'));
   timeStep = str2double(get(handles.tbTimeStep, 'string'));
   if (chosenScenario == 0)
      msgbox('Error: No scenario chosen. Please choose a scenario by selecting
the corresponding radio button');
      return;
   end

   if (secLength <= 0.0)
      msgbox('Error: Invalid link length. Please put a valid link length for
computation (>0)');
      return;
   end

   if (timeStep <= 0.0)
      msgbox('Error: Invlid time step value. Please put a valid time step for
computation (>0)');
      return;
   end

   switch chosenScenario
      case 1
         lwr_ts1(timeStep, secLength);
         return;
      case 2
         lwr_ts2(timeStep, secLength);
         return;
      case 3
         lwr_ts3(timeStep, secLength);
         return;
      case 4
```

```
        lwr_mod_ts1(timeStep, secLength);
        return;
    end
    %;


function tbSecLen_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbSecLen (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function tbSecLen_Callback(hObject, eventdata, handles)
% hObject    handle to tbSecLen (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbSecLen as text
%        str2double(get(hObject,'String')) returns contents of tbSecLen as a double


% --- Executes during object creation, after setting all properties.
function tbTimeStep_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbTimeStep (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
function tbTimeStep_Callback(hObject, eventdata, handles)
% hObject    handle to tbTimeStep (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbTimeStep as text
%        str2double(get(hObject,'String')) returns contents of tbTimeStep as a
double


% --- Executes during object creation, after setting all properties.
function tbHelpHeader_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbHelpHeader (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function tbHelpHeader_Callback(hObject, eventdata, handles)
% hObject    handle to tbHelpHeader (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbHelpHeader as text
%        str2double(get(hObject,'String')) returns contents of tbHelpHeader as a
double


% --- Executes on button press in bCloseHelp.
function bCloseHelp_Callback(hObject, eventdata, handles)
% hObject    handle to bCloseHelp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% --- Executes during object creation, after setting all properties.
function tbHelpText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbHelpText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function tbHelpText_Callback(hObject, eventdata, handles)

function enableHelpFrame(handles)
   % get the attributes of the help message. A help message consists of
   % two attributes: header and the text message.
   heading = handles.msg_header;
   message = handles.msg_message;
   msgbox(heading, message);

% This function opens up the help screen for modified scenario 1.
function bHelpSc1Mod_Callback(hObject, eventdata, handles)
```

LWR.m

```
% FUNCTION HEADER -----
% Function Overview :   This function is the starting point to run the
% actual simulation. It reads the various parameters from the property file
% and then use then appropriately to compute the vehicular density.
% input parameter(s):
%       sectionLen  :   The link length.
%       timeStep    :   The time step between each subsequent observation.
% ouput parameter(s):   NONE
% Creater           :   Dr. Paul Nelson     Initial Creation
%                       Nishant:            07/23/03    Initial creation
%                       Nishant:            06/10/04    Added comments.
% FUNCTION HEADER -----

function lwr_ts2(timeStep, sectionLen)
```

```
% The following global variables are defined in the property file(sc2_prop)
global TSM DIN SOUT INITIAL FLOW UP_BNDRY DOWN_BNDRY
TM_START TM_END HEADING x_matrix

% All the global variables are initialized
[TSM DIN SOUT INITIAL FLOW UP_BNDRY DOWN_BNDRY TM_START
TM_END HEADING] = sc2_prop;

% calculate the number of time steps and numbers of sections
numTimeSteps = (TM_END - TM_START)/ timeStep;
numSections = ( DOWN_BNDRY - UP_BNDRY)/sectionLen;

% The following section sets up spatial and temporal observation points.
x_matrix = linspace(UP_BNDRY+sectionLen/2, DOWN_BNDRY-sectionLen/2,
numSections);
t_matrix = linspace(TM_START, TM_END, numTimeSteps);

% This computes the initial density condition of the roadway under
% consideration.
k(:,1) = feval(INITIAL,numSections)';

% The following loop calculates the vehicular density at the (x,t)
% observation points.
for timeCtr = 2:numTimeSteps+1
   % Calculate the inflow and outflow from the entry link
   inflow = feval(FLOW,-1,k(1,timeCtr-1),0,x_matrix(1),t_matrix(timeCtr-1));
   outflow = feval(FLOW,k(1,timeCtr-1),k(2,timeCtr-
1),x_matrix(1),x_matrix(2),t_matrix(timeCtr-1));
   % Update concentration at upstream entry subsection
   k(1,timeCtr) = k(1,timeCtr-1) + (timeStep/sectionLen)*(inflow-outflow);
   % Update concentration in interior subsections
   for secCtr = 2:numSections-1
      % inflow to subsection secCtr = outflow from subsection secCtr-1
      inflow = outflow;
      outflow = feval(FLOW,k(secCtr,timeCtr-1),k(secCtr+1,timeCtr-
1),x_matrix(secCtr),x_matrix(secCtr+1),t_matrix(timeCtr-1));
      k(secCtr,timeCtr) =  k(secCtr,timeCtr-1) + (timeStep/sectionLen)*(inflow-
outflow);
   end
   inflow = outflow;
   % Outflow from downstream exit subsection
   outflow = feval(FLOW,k(numSections,timeCtr-1),-
1,x_matrix(numSections),0,t_matrix(timeCtr-1));
```

```
   k(numSections,timeCtr) = k(numSections,timeCtr-1) +
(timeStep/sectionLen)*(inflow-outflow);
end

% This section reduces the observation points. This makes the plot easier
% to plot.

JT=10;
taxis=linspace(0,numTimeSteps*timeStep,numTimeSteps/JT+1);% Plot time
mesh
for timeCtr=0:numTimeSteps/JT
  for secCtr = 1:numSections
      conc1(secCtr,timeCtr+1)=k(secCtr,timeCtr*JT+1);
   end
end

% This actually creates the 3-D result
figure('name',HEADING);
mesh(x_matrix,taxis,conc1')
xlabel('distance (miles)')
ylabel('time (hours)')
zlabel('density (vpmpl)')
title('Freeflow Example, Upstream Method')
view(35,75)
```

Godunov_ts2.m

```
% FUNCTION HEADER -----
% Function Overview :   This function sets up the various functions involved
%                in getting the outflow from the link in question.
% input parameter(s):
%          kup     : vehicular density at the upstream end of the link
%          kdown   : vehicular density at the downstream end of the link
%          xup     : x-coordinate of the upstream end
%          xdown   : x-coordinate of the downstream end
%          presTime: time at which the computation is being done
% ouput parameter(s):
%          q       : Flow from the link under consideration
% Creater          :   Dr. Paul Nelson     Initial Creation
%                   Nishant:          07/23/03    Initial creation
%                   Nishant:          06/10/04    Added comments.

function q = Godunov(kup,kdown,xup,xdown,presTime)
global TSM DIN SOUT
```

```
% The following is Godunov, per p. 664 of Lebacque, Procs. ISTTT 13
% If it is the entry link, then the demand is defined by DIN. If it is the
% exit link then the supply is defined by SOUT. If it is any other link,
% then the supply and demand are calculated using the demand and supply
% function.
if kup == -1
   dup = feval(DIN,presTime);
   sdown = supply_ts2(kdown,xdown,presTime);
elseif kdown == -1
   sdown = feval(SOUT,presTime);
   dup = demand_ts2(kup,xup,presTime);
else
   dup = demand_ts2(kup,xup,presTime);
   sdown = supply_ts2(kdown,xdown,presTime);
end
q = min(dup,sdown);
```

triangular_ts2.m

```
% FUNCTION HEADER -----
% Function Overview :   % This is a triangular traffic stream module designed to
replace
% the Greenshields Traffic Stream Model. In the ross scenario 2, the
% capacity of lanes changes with X. Initially we have three lanes which
% then converge into 2 lanes and then into one lane. Hence, the maximum
% allowed density also changes.
% Input parameters:
%   K: density at the given X
%   presX: the x-coordinate
%   presTime : the present time
%
% Output parameters
%   Kcap: Density at which maximum capacity flow occurs. (2000/ max. free
%   speed)
%   Qcap: Maximum capacity flow
%   Q   : Flow
% Creater          :   Dr. Paul Nelson     Initial Creation
%                  Nishant:          07/23/03    Initial creation
%                  Nishant:          06/10/04    Added comments.
% FUNCTION HEADER -----

function [Kcap,Qcap,Q]=triangular(K,presX,J0)
```

```
vmax=63; %mph
Qcap=2000; %vpmpl
Kcap=Qcap/vmax;
Kjam=143;
Xlimit1 = 5; % limit at which the first lane drop occurs
Xlimit2 = 9; % limit at which the second lane drop occurs
Xlimit3 = 13; % end of the roadway.
if (presX >= 0 & presX <= Xlimit1)
   Kcap=3*Kcap;
   Qcap=3*Qcap;
   Kjam=3*Kjam;
elseif (presX > Xlimit1 & presX <= Xlimit2)
   Kcap=2*Kcap;
   Qcap=2*Qcap;
   Kjam=2*Kjam;
end

% Due to the 'incident' happening between 4.9 miles and 5.3 miles, the free
% flow speed decreases from 63 to 50 and leads to change in the triangular
% equation.
%if ( presX >= 4.9 & presX <= 5.3 )
 %   vmax = 50;
  %  Kcap = Qcap/vmax;
  %end

if (K <= Kcap)
   Q = K*vmax;
else
   Q=Qcap*(Kjam-K)/(Kjam-Kcap);
end
```

sc2_prop.m

```
% FUNCTION HEADER -----
% Function Overview :   This is the property file which defines the various
% parameters for ross scenario 2.
% input parameter(s):
% ouput parameter(s):
%       TSM         : This is the file/ function containing the traffic stream
%                 model.
%       DIN         : This defines the demand at the entry link.
%       SOUT        : This defines the supply at the exit link.
%       INITIAL     : This defines the initial density along the roadway.
%       FLOW        : This defines the file/ function containing the
```

```
%               implementation to compute flow.
%       UP_BNDRY    : The upstream boundary co-ordinate.
%       DOWN_BNDRY  : The downstream boundary co-ordinate.
%       TM_START    : The simulation start time.
%       TM_END      : The simulation end time.
%       HEADING     : The heading shown on the result

% Creater         :   Nishant:          07/23/03    Initial creation
%                     Nishant:          06/10/04    Added comments.
% FUNCTION HEADER -----
function [TSM, DIN, SOUT, INITIAL, FLOW, UP_BNDRY, DOWN_BNDRY,
TM_START, TM_END, HEADING] = sc2_prop

TSM ='triangular_ts2'
DIN = 'din_ts2'
SOUT = 'sout_ts2'
INITIAL = 'initial_ts2'
FLOW = 'godunov_ts2'
UP_BNDRY = 0
DOWN_BNDRY = 13
TM_START = 0
TM_END = 1
HEADING = 'Ross Scenario 2'
```

Initial_ts2.m

```
% FUNCTION HEADER -----
% Function Overview :   This function calculates Initial density for Traffic
Scenario 2.
% input parameter(s):
%         I0      :   The number of links.
% ouput parameter(s):
%       supply      :   The supply at the link.
% Creater         :   Dr. Paul Nelson     Initial Creation
%                     Nishant:          07/23/03    Initial creation
%                     Nishant:          06/10/04    Added comments.
% FUNCTION HEADER -----%
function K = Initial_ts0(I0)
global x
for i = 1:I0
   K(i) = 800/63;
end
```

supply_ts2.m

```
% FUNCTION HEADER -----
% Function Overview :   This function calculates the supply at the link.
% input parameter(s):
%          kdown   : vehicular density at the downstream link
%          xdown   : x-coordinate of the upstream end
%          j0      ; Time.
% ouput parameter(s):
%       supply      :   The supply at the link.
% Creater          :   Dr. Paul Nelson     Initial Creation
%                  Nishant:          07/23/03    Initial creation
%                  Nishant:          06/10/04    Added comments.
% FUNCTION HEADER -----

function sdown = supply(kdown,xdown,j0)
global TSM DIN SOUT

[kcap,qcap,qdown] = feval(TSM,kdown,xdown,j0);
if kdown <= kcap
   sdown = qcap;
else
   sdown = qdown;
end
```

demand_ts2.m

```
% FUNCTION HEADER -----
% Function Overview :   This function calculates the demand at the link.
% input parameter(s):
%          kup     : vehicular density at the upstream end of the link
%          xup     : x-coordinate of the upstream end
%          j0      ; Time.
% ouput parameter(s):
%       demand      :   The demand at the entrant link.
% Creater          :   Dr. Paul Nelson     Initial Creation
%                  Nishant:          07/23/03    Initial creation
%                  Nishant:          06/10/04    Added comments.
% FUNCTION HEADER -----

function dup = demand(kup,xup,j0)
global TSM DIN SOUT

[kcap,qcap,qup] = feval(TSM,kup,xup,j0);
if kup <= kcap
```

```
   dup = qup;
else
   dup = qcap;
end
```

sout_ts2.m

```
% FUNCTION HEADER -----
% Function Overview :   This function specifies the supply at the exit
% link as a function of time.
% input parameter(s):
%       presTime    :   The current time.
% ouput parameter(s):
%       demand      :   The demand at the entrant link.
% Creater          :   Dr. Paul Nelson     Initial Creation
%                  Nishant:           07/23/03    Initial creation
%                  Nishant:           06/10/04    Added comments.

function sout = Sout_qr(presTime)
   sout = 2000;
```

din_ts2.m

```
% FUNCTION HEADER -----
% Function Overview :   This function specifies the demand at the entry
% link as a function of time.
% input parameter(s):
%       presTime    :   The current time.
% ouput parameter(s):
%       demand      :   The demand at the entrant link.
% Creater          :   Dr. Paul Nelson     Initial Creation
%                  Nishant:           07/23/03    Initial creation
%                  Nishant:           06/10/04    Added comments.
% FUNCTION HEADER -----

function demand=DIN_QR(presTime)

if (presTime >= 0.0 & presTime <= 0.1)
   demand = 1800;
elseif (presTime > 0.1 & presTime <= 0.2)
   demand = 2300;
elseif (presTime > 0.2 & presTime <= 0.3)
   demand = 2800;
elseif (presTime > 0.3 & presTime <= 0.5)
```

```
   demand = 3300;
elseif (presTime > 0.5 & presTime <= 2.0)
   demand = 5800;
end
```

VITA

Nishant Kumar

4306 156th Avenue NE, Apt # TT 258, Redmond, WA – 98052

Ph: 425-301-4086, E-mail: iNishant@gmail.com

EDUCATION:

- Texas A&M University**,** College Station, Texas
  Master of Science (Computer Science) Graduation Date: Aug 2004.
- Indian Institute of Technology, Bombay, INDIA.
  Bachelor of Technology (Civil Engineering) Graduation Date: May 1998.

WORK EXPERINCE:

- **Texas Transportation Institute, Texas A&M University System**.
  Graduate Research Assistant. September 2002-Present
- **Infosys Technologies Ltd., Bangalore**
  Programmer Analyst. June 1998-July 2002.

PAPERS:

- *P. Nelson, N. Kumar.* Point Constriction, Interface and Boundary Conditions for Kinematic-wave Model. *Transportation Research Board.* URL: http://www.trb.org/news/blurb_detail.asp?id=1121. Last Modified: 01/12/2004.